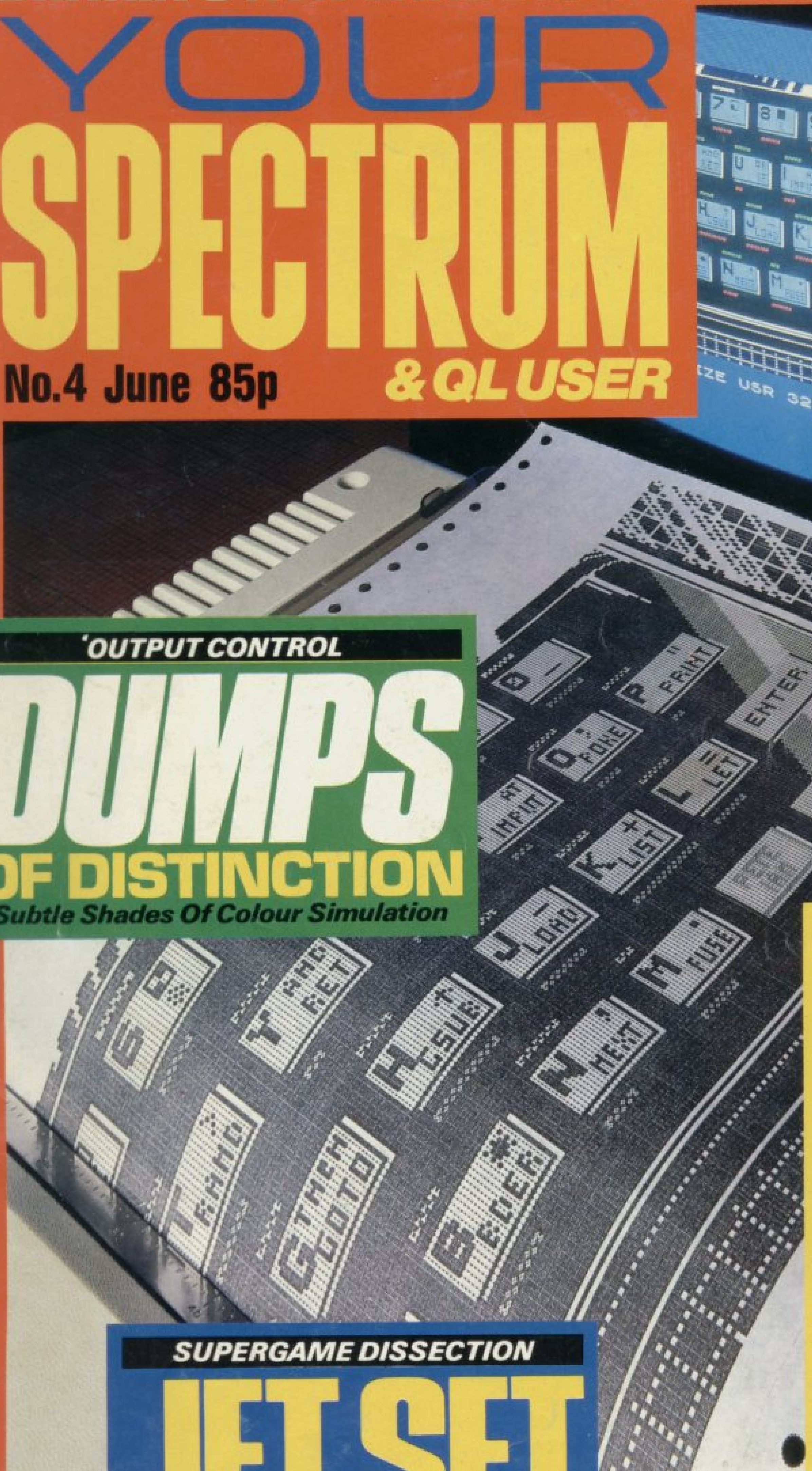
BRITAIN'S NO.1 WINDOW ON THE SPECCY WORLD!

YOUR SPECTRUM & QL USER

No.4 June 85p



'OUTPUT CONTROL

DUMPS OF DISTINCTION

Subtle Shades Of Colour Simulation

INSIDE QL USER



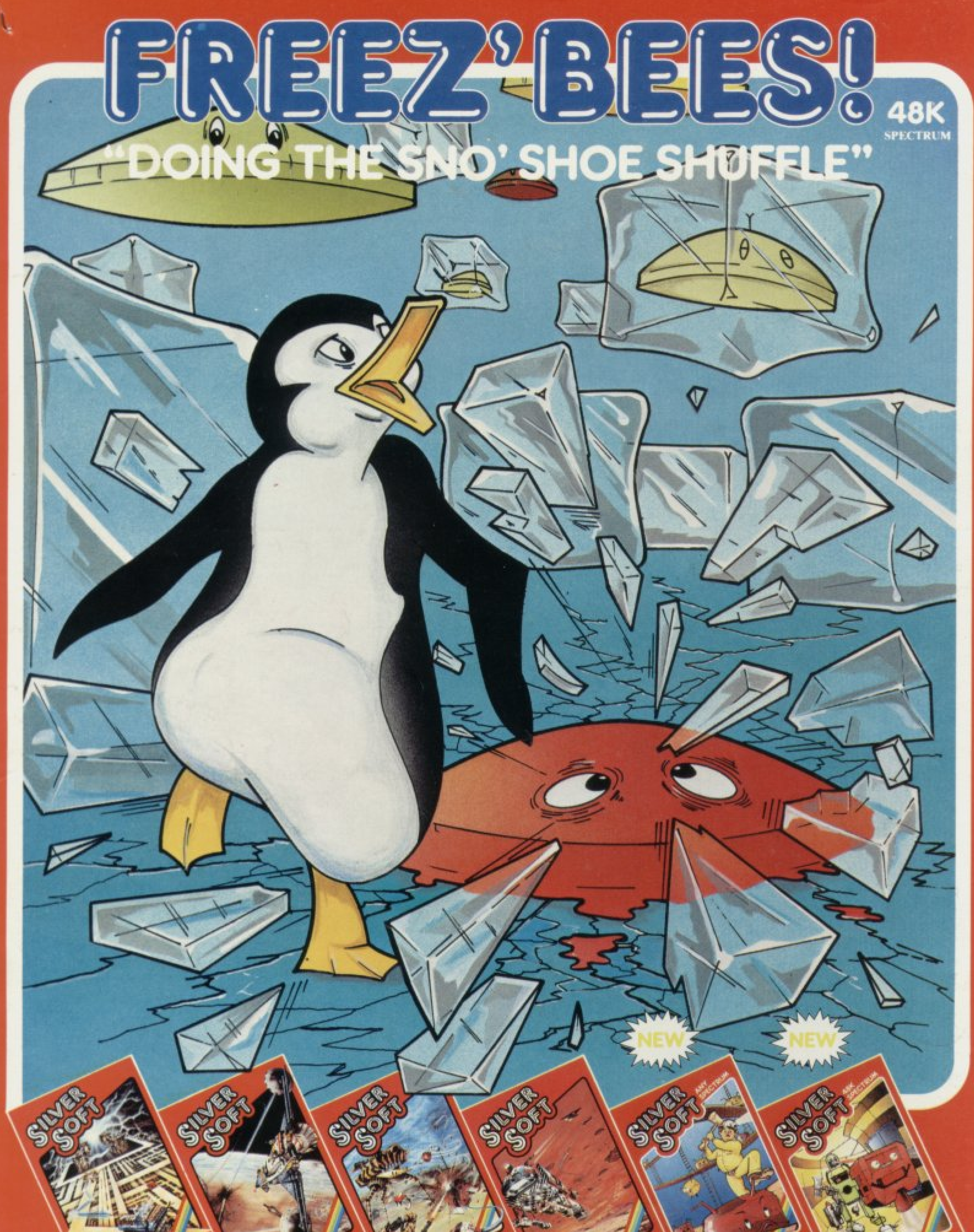
A CLOSE ENCOUNTER

. . . Of The Testbench Kind

SUPERGAME DISSECTION

JET SET WILLY

Confessions Of A Closet Hacker



FREEZ'BEES!
48K
SPECTRUM
"DOING THE SNO' SHOE SHUFFLE"
NEW
NEW
SILVER SOFT
BRAIN DAMAGE
ARMAGEDDON
EXTERMINATOR
MISSION IMPOSSIBLE
SAM SPADE
ROBOT RIOT

INSIDE YS

FEATURES

ON THE BENCH

QL USER



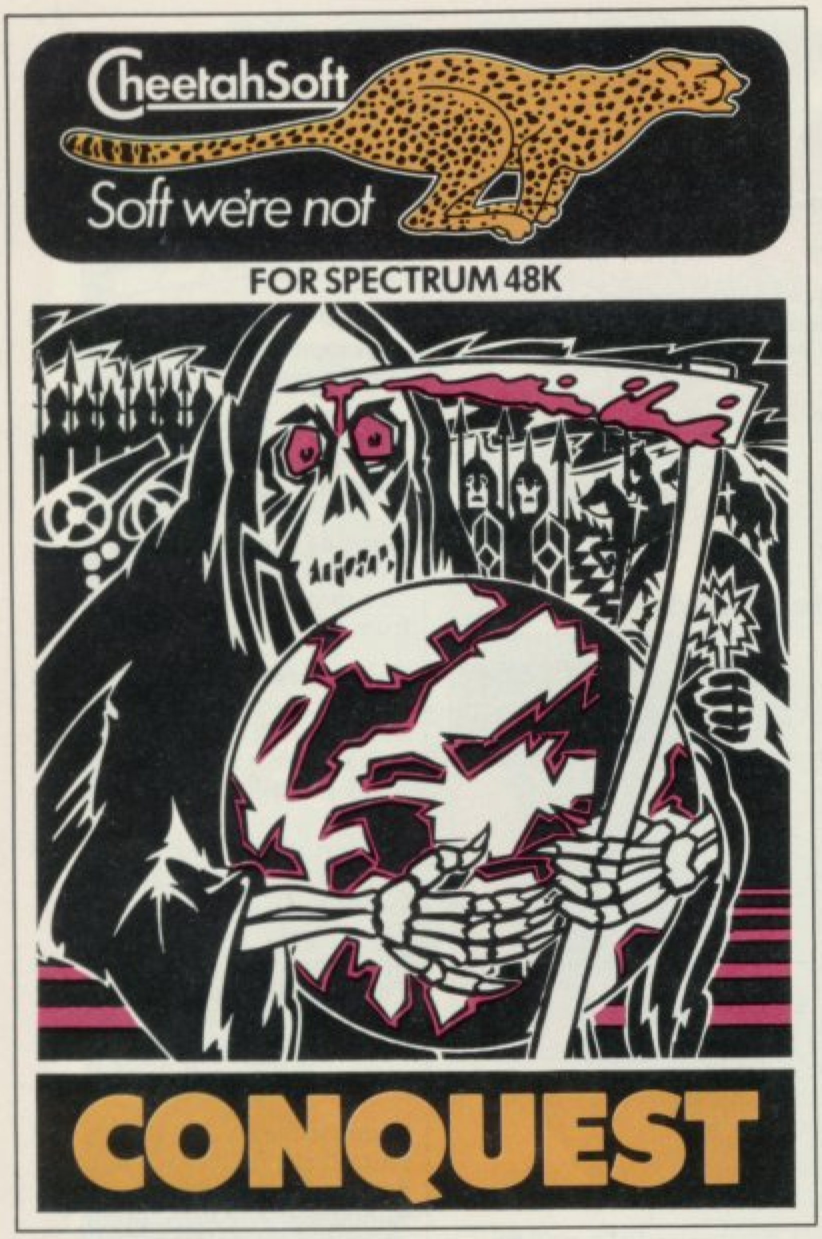
REGULARS



Cover photography by Ian McKinnell

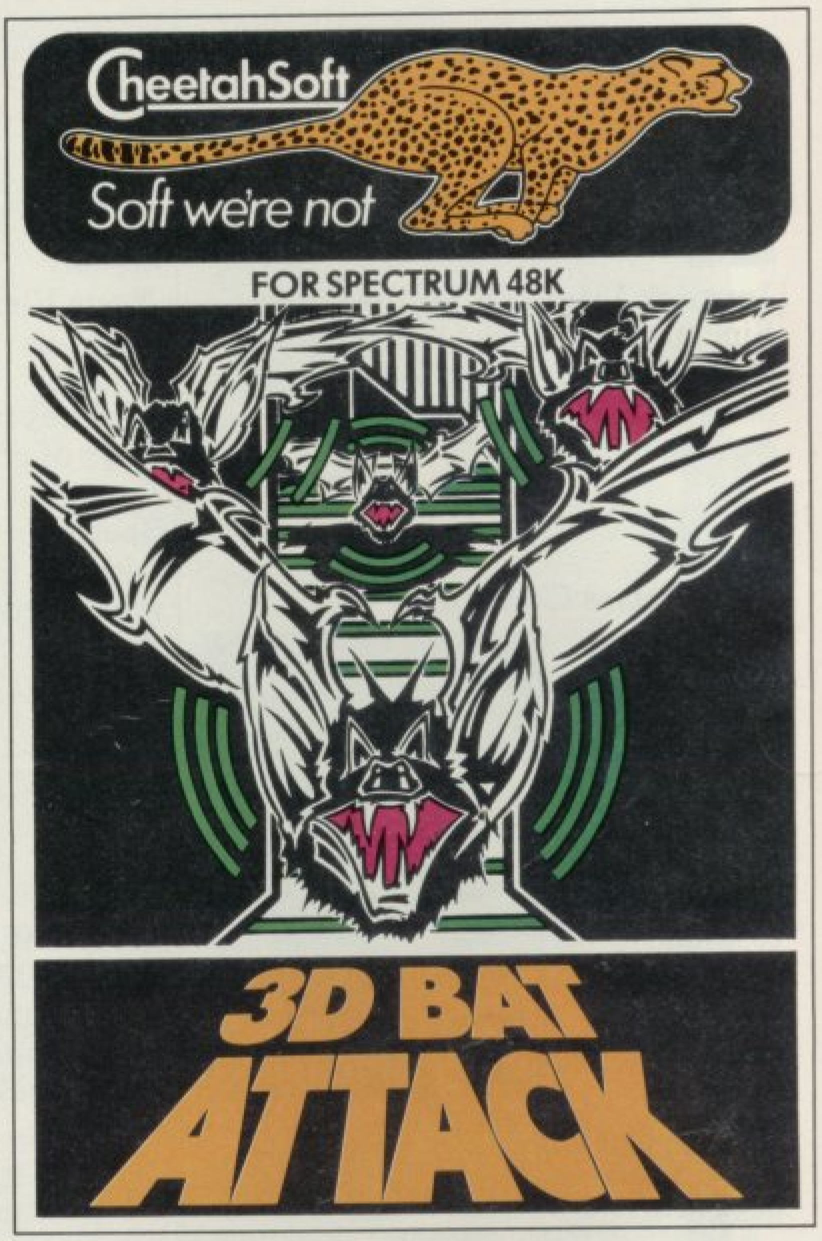
Editor Roger Munford; **Managing Editor** Bruce Sawford; **Deputy Editor** Tina Boylan; **Editorial Assistant** Peter Shaw; **Editorial Consultant** Andrew Pennell; **Software Consultant** Gavin Monk; **Contributors** Ian Beardsmore, Ron Smith, Stephen Adams, Damir Skrgatic, Simon Goodwin, Toni Baker, Peter Jackson, Dilwyn Jones, Paul Walton, Andrew Pennell, Max Phillips; **Art Editors** Jimmy Egerton, Hazel Bennington; **Art Assistant** Steve Broadhurst; **Group Advertisement Manager** Jeff Raggett; **Advertisement Manager** Shane Campbell; **Advertisement Executive** Jason Wood; **Typesetting Manager** Derek Cohen; **Typesetters** Beverley Douglas, Maggie Kayley, Velma Miller; **Production Manager** Sonia Hunt; **Group Art Director** Perry Neville; **Publisher** Steve England; **Distribution Manager** Colin James; **Published by** Sportscene Specialist Press Ltd, 14 Rathbone Place, London W1P 1DE. Company registered in England. **Telephone** (all departments) 01-631 1433. **Telex** 8954139 BunchG. **Reproduction** Graphic Ideas, London; **Printed by** Chase Webb Offset, St Austell, Cornwall; **Distribution** Seymour Press, 334 Brixton Road, London SW9. Telephone 01-733 4444.

Your Spectrum is a monthly publication.



3D BAT ATTACK An all action, 3 dimensional maze game where you gather up blocks of gold, at the same time pitting your wits against vicious vampire bats whose only purpose in life is to locate, hunt and kill you.

4 levels of skill. At each level the game gets faster and more complicated, and the vampires more dangerous.



CONQUEST A tactical game which even veteran players will find both challenging and rewarding.

Mediterranean Europe is divided into grid squares. Your aim, as Emperor, is to gain 100 squares of territory as quickly as possible – at the same time dealing with Barbarian counter-attacks, plagues, civil war and rival Emperors.

5 levels of skill, plus a continuous 'Conquest' game where all the difficulty levels are thrown in together.

AFTER THESE, THE REST IS KIDS STUFF.

Are you ready for CheetahSoft? There's only one way to find out. But be warned: these vampire bats know a good meal when they see one. And our friend with the scythe has had years of experience . . .

So don't play unless you're ready to play the game for real. Because you'll find there's one sure thing about CheetahSoft: Soft we're not.

£6.95 at all good stores.

CheetahSoft

Soft we're not

RAINBOW'S SPECTRUM PROGRAMMABLE JOYSTICK INTERFACE WITH SOUND

NOW IMPROVED

* ***The keyboard is NOT disabled when using the Joystick,***

* ***You can have a Through Port** – See Below.*

As before, its features are:

PROGRAMMABLE — Allows any joystick position to represent any key on the Board – *without* wires, or leads, or tapes!

UNIVERSAL — Enables use of *ALL* Software

AMPLIFIED SOUND — Internal micro-chip amplifier boosts Spectrum's beep with simple connection to ear or mic socket.

EASY TO USE — All joystick positions are programmed simply by selecting joystick position, pushing desired key on spectrum, releasing joystick, and then the key.

ADAPTABLE — Accepts diagonal positions.

ONLY £24
Extra for Through Port £3

plus £1 p&p.



This is a top quality product with guarantee

Send Cheque or P.O. to: **Rainbow Electronics**

Glebe House, South Leigh, Witney, Oxfordshire OX8 6XJ
Tel. Witney (0993) 5432

Y.S.

Friendly Face

MICRODRIVE UTILITY

ADDS microdrive commands to basic

MERGES with Masterfile & Tasword Two

COPIES operating system to cartridges

MONITOR

P.O. Box 442 London NW7 2JF
Phone: 01-959 1787
Cable: MONITOR London NW7

Cassette £7·95
Cartridge £14·95

ACTION IN THE THIRD REICH...



SPECIAL OPERATIONS

NEW

This graphic adventure allows you to pick your own specialist team. Evade the German patrols in order to locate, photograph, penetrate and eventually destroy a germ warfare research centre, stealing the top secret cultures on the way.

Spectrum 48K £5.95

THE THIRD DYNASTY...

NEW

MUMMY MUMMY



Woken from eternity by a mad archeologist and armed only with a spade you must bury the spirits in the masonry of the Pyramid if you are ever to read the cartouches and return to your golden sarcophagus.

Spectrum 48K £5.95

OR THE THIRD DIMENSION.

NEW

GRID PATROL



Fast and furious. Eight different screens test your lightning reactions as you attempt to wipe out the aliens in a devastating crossfire from surrounding tanks. High score table. 100% machine code.

Any Spectrum £5.95

JOYSTICK COMPATIBLE KEMPSTON AND CURSOR JOYSTICKS

Simply fill in the coupon below and POST TODAY!

ORDER NOW!

GSH

ACTIONMASTER GAMES - AVAILABLE AT SELECTED OUTLETS OF:- JOHN MENZIES, NSS, CO-OP, SPECTRUM, MAKRO, TIGER TRADERS, LIGHTNING DEALERS & BETTER RECORD SHOPS. SELECTED TITLES AT W.H.SMITH

You'll also find the full range of Actionmaster, Adventuremaster and Warmaster games in our free colour catalogue.

Special Operations	48K Spectrum £5.95 [ ]	Two-Gun Turtle	Any Spectrum £5.95 [ ]
Mummy Mummy	48K Spectrum £5.95 [ ]	Bedlam Blaster	Any Spectrum £5.95 [ ]
Grid Patrol	Any Spectrum £5.95 [ ]	Beetlemania	Any Spectrum £5.50 [ ]
Micromouse	Any Spectrum £5.95 [ ]	Please send me my FREE colour catalogue [ ]	

I enclose a cheque/P.O. for £ ______ made payable to M C Lothlorien.

Please debit my Access A/C No. ___/___/___/___

Name ______

Address ______

· SU 5/84

Post to: M C Lothlorien, 56a Park Lane, Poynton, Stockport, Cheshire. SK12 1RE
Tel: Poynton (0625) 876642

LOTHLORIEN

more action for your money

FRONTLINES

FAIR PLAY

Many feel that one of the best exhibitions around is the Computer Fair that's held at Earl's Court each year, and 1984's effort should be no exception. It always gets pretty jam-packed, but this time the organisers are doing a PCW and dividing it into two distinct sections — home and business. Not only that, they've opted for a 'Business/ Trade Only' day so that 'serious' people won't have to contend with the expected army of enthusiasts (the organiser claimed, all told, an amazing 50,000 for last year's bash).

Sinclair Research will, of course, be there (after all that's where the Spectrum was unveiled in 1982) and this year we may even catch a sight of that legendary beast, the Abominable QL. Other than that there'll be all the usual Sinclair Research bits and bobs — the Speccy, ZX Microdrives, Interfaces 1 and 2, ZX Printer, and a wide range of software cassettes. Sinclair Research will be gleefully taking orders

The Computer Fair at London's Earl's Court will be open from 10.00am — 6.00pm on June 14th (Business/Trade only), then June 15th and 16th (same times) and from 10.00am — 5.00pm on June 17th.

EXPANDABLE MOTHER

Currah has delved yet further into hardware with the release of an expandable motherboard.

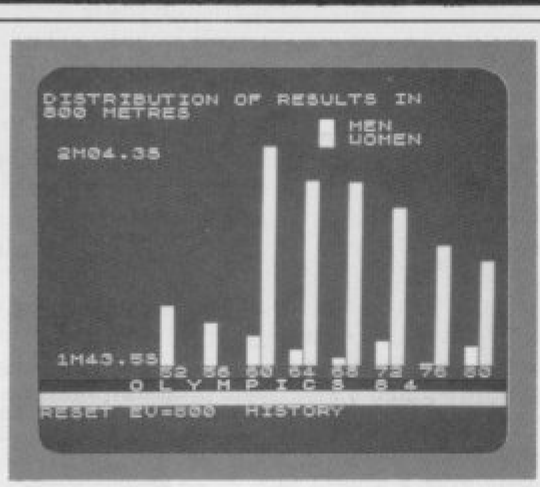
It's called the MicroSlot and Currah say that it has two adjacent slots which allow for easy connection of any chosen combination of accessories — such as a printer connector, ZX Interface 2 or even a Currah MicroSpeech. By stacking MicroSlots in a row, the Speccy system can be expanded as far as the power supply will let you.

The 'Spectrum black' unit comes as a prism-shaped box and will be available either direct from Currah, or through national retail outlets — price £14.95.

OLYMPIAN FEATS

Believe it or not, there really are people who sit around waiting in breathless anticipation of the next Olympic Games. And if you are one, then have we got news for you! Because now, not only will there be *hours and hours* of it on the telly, but you can also gen up on all the statistics with the help of your Speccy.

Storm Software has brought out a double cassette *Olympics 84* package to coincide with this summer's goings-on in Los Angeles. It contains a computer database of Olympic results, facts and figures, including all medal winners from 1896 to 1980 — by country and with all their winning scores. This means your average sportsnut will be able to examine all the results . . . by year, individual sport or event, by Gold, Silver or Bronze medal winner or by any combination of the above. There's also a History option which illustrates how the results in an event have improved over the years, and it encourages males to commit chauvenistic suicide by allowing one to compare the relative performances of women against men.



Historical results are compared by year and number of men and women. Other comparisons can be made by country and winning scores. This also makes it easy to see how results have improved over the years.

So, if you think that Coe, Ovett and Cram are set to excell, then perhaps you'd like to take a look at when Great Britain last won three Olympic gold medals in middle distance running or even when it won three gold medals in any athletics event. With this package, all is possible.

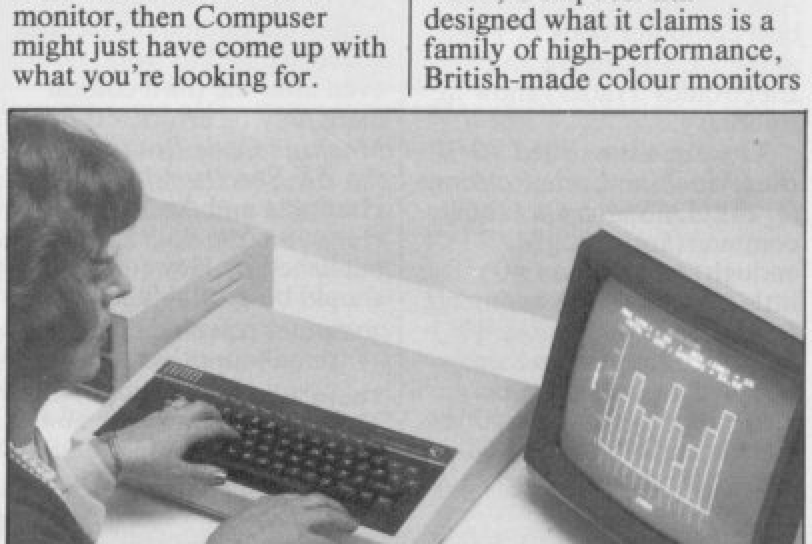
Included with the two cassettes is a booklet which gives an amusing anecdotal account of the world's most famous 'shamateur' games event — from 1896 in Athens through to Moscow 1980. And a further bonus is that with the Input cassette you can add the names, nationalities and times of all the 1984 medal winners and compare them with those of 1980 and 1976.

For more words on this enterprising package, contact Storm Software on 0935 813528.

PRETTIER PICTURES

If all you ever wanted for Christmas was a decent monitor, then Compuser might just have come up with what you're looking for.

New-ish to the computer scene, Compuser has designed what it claims is a family of high-performance, British-made colour monitors to suit all pockets and demands.



Compuser's British-made colour monitors are reasonably priced and can be supplied with carrying handles on request.

The 14-inch standard resolution composite monitor has two inputs, RGB/TTL and PAL encoded composite signal, with separate sound outputs which make it compatible with any computer that has RGB monitor output or composite signal output — so it's capable of being driven from a Speccy either through an interface unit or directly via a simple modification. It can be used with a VCR, to display either tape or TV picture. The monitors are supplied in attractive cabinets (it says here) and can be supplied with carrying handles on request.

The standard resolution monitor costs £265 plus VAT and is available direct from Compuser Ltd. You can phone the company on 0689 46116.

BEYOND SOFTWARE

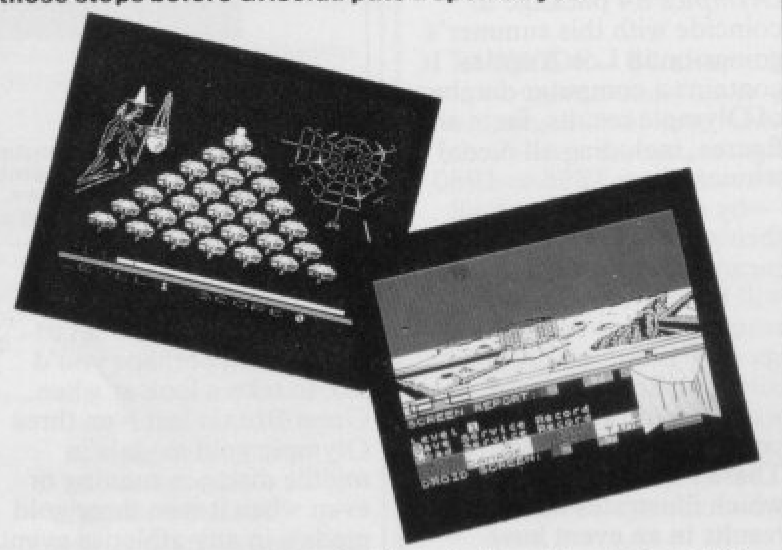
Beyond, has just launched three new games for the liddle 'ol Speccy, the first of which it reckons will take adventure players by storm.

It's called *Lords of Midnight* and, according to the description, it's not actually an adventure at all, but — rather — an 'Epic'. The idea is that, unlike other adventures, you don't need to restrict yourself to just one character and your routes are no longer dictated by the programmer in advance. *This* program knows where a given character is on the map, checks the direction he is facing and draws the whole scene, foreground to background, in perspective. Thus, every time you leave a location and return to it from a new direction, it'll be drawn again with a different background; and every time you move forward, distant peaks become mighty mountains and houses grow into great towers, flanked by forests. This is all made possible through (it's claimed) a new programming technique called landscaping which gives the program a possible 32,000 views.

The idea of the game is to gather armies from the various people you meet; you have to convince them to join you, but of course there are some who are simply enemies and must be destroyed. As the force splits to defeat the evil Witchking Doomark you can see the dramatic happenings through the eyes of each character. There's an accompanying 32-page booklet which includes a map of the land and the game should retail for around £9.95.

The second game from Beyond is called *Psytron*, and in it you play the part of a computer (Psytron itself) which controls the massive Betula 5 installation — a sort of moonbase type set-up. The computer controls every aspect of the place, from defending the base against evil intruders to assessing everyone's oxygen needs and allocating food and work. The problem is that the base is being constantly attacked and bombed from the skies while saboteurs (of the three-legged canine variety) run loose underneath the floorboards. The aim of the game is to keep the base running, and all human life is expendable in order to maintain the mighty Psytron.

In Spellbound you must use your frog's legs to hop fast down those steps before Griselda puts a spell on you!



Psytron's view of the Betula 5 installation. There's terror in the skies and under the floorboards.

There are six levels and, according to the accompanying booklet, they've been carefully designed to take you through the game step-by-step; you musn't overload your all-too-human mind.

You may be interested to know that level six has only one aim — to survive for half an hour! Beyond is challenging players to achieve this great feat and the reward is . . . wait for it . . . a QL. Obviously, the company's not expecting anyone to manage it for a while yet. Anyway, full details are in the pack. The game, by the way, sells for £7.95.

Last in the bunch comes *Spellbound*, where you take the part of a frog and attempt to race down the steps of the fortress before Griselda the witch catches you with her spells. Straightforward stuff it may be, but there are 12 levels of action to get through. *Spellbound* costs £5.95 and all Beyond's games should be available in your shops now.

COOLING THE COSTS

Is your home so cold during the winter that you don't just suffer with frozen pipes and frost on the inside of your windows, but you actually have to defrost the dog in the mornings? Yes? Then maybe you'd be interested in a program from Brane Software that's snappily entitled *Cut Your Heating Bills*. It can calculate savings from double glazing, cavity wall insulation, wall, floor and roof insulation, and change of fuel, heater, ventilation or thermostat setting. It can also compare true costs of different fuels and you can enter any combination of different insulation materials; results can be saved on tape or just printed out.

The program runs on the 48K Spectrum and is available through mail order and costs £7.95 from Brane Software. Phone them on 0736 72562.

HEWSON'S BEST BOOK

Hewson Consultants, best known for its simulation software, picked up the CTA (Computer Traders Association) Award for the best computer book of 1983 at the LET (Leisure and Electronics Trader) show in February.

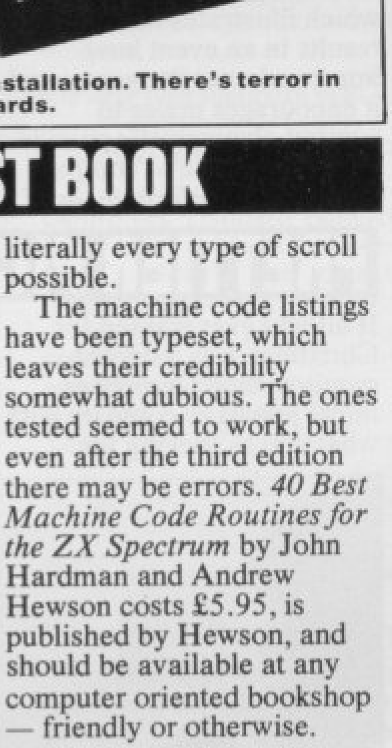
The tome is entitled *40 Best Machine Code Routines for the ZX Spectrum*, and it contains (surprisingly enough) no less than 40 nifty little machine code oddments for the Speccy.

But it doesn't stop there, because the 144-page book also explains what machine code is and why you should use it —and how a Speccy's mind works. Now into its third edition, apparently the publication is still selling well.

All the routines featured are comprehensively documented with plenty of instructions on how to get things running. Some of the best include a renumber routine which handles GO TO, GO SUBS, etc., and literally every type of scroll possible.

The machine code listings have been typeset, which leaves their credibility somewhat dubious. The ones tested seemed to work, but even after the third edition there may be errors. *40 Best Machine Code Routines for the ZX Spectrum* by John Hardman and Andrew Hewson costs £5.95, is published by Hewson, and should be available at any computer oriented bookshop — friendly or otherwise.

There's 144 pages to tell you all about machine code and how to use it.



SOFT SUPREMOS

Websters Software has been appointed the first ever software-only distributor for Sinclair Research in the UK. The company has only been in business for the last 12 months, but it's already moved fast at building up a wide base of retail accounts. Anton Boyes, retail manager of Sinclair Research, says that "Software is an increasingly important part of our business. We believe that Websters will help to extend and improve the distribution of our products to retailers throughout the country."

This can only improve matters for those of us who want to get hold of the stuff, and Websters seem over-the-moon with the idea. It's MD was heard of remark: "Sinclair Research are world leaders in the home computer market and one couldn't ask for a more significant vote of confidence. This is good news!" Websters Software is on 0483 62222.

FORESTRY COMMISSION

Calling all *Forest* devotees, Have you been amazed to discover that yo never run off the edge of the action although *your* map is only 2Km by 2Km? Then grab pen and paper and immediately send off for your *Complex Forest* map from Phipps Associates, 172 Kingston Road, Ewell, Surrey (or phone 01-393 0283).



PARK IT IN THE ROM

Parker has moved into the home micro market with cartridges for use on the Interface 2.

Three of the five releases will be translations of arcade games: *Popeye, Q*Bert* and *Return of the Jedi — Death Star Battle*. Two others are completely new. Fresh from the arcades, *Gyruss* is a shoot 'em up with a new perspective. The pilot ship travels 360 degrees around the edge of the screen and the opposing marauders attack from the centre.

The final release is *Star Wars — The Arcade Game* which graphically re-enacts the destruction of the Death Star — taken, of course, direct from the original film.

Parker hopes to have the games finished by June/July for release in August.

NEW 'FACE FOR THE SPECCY

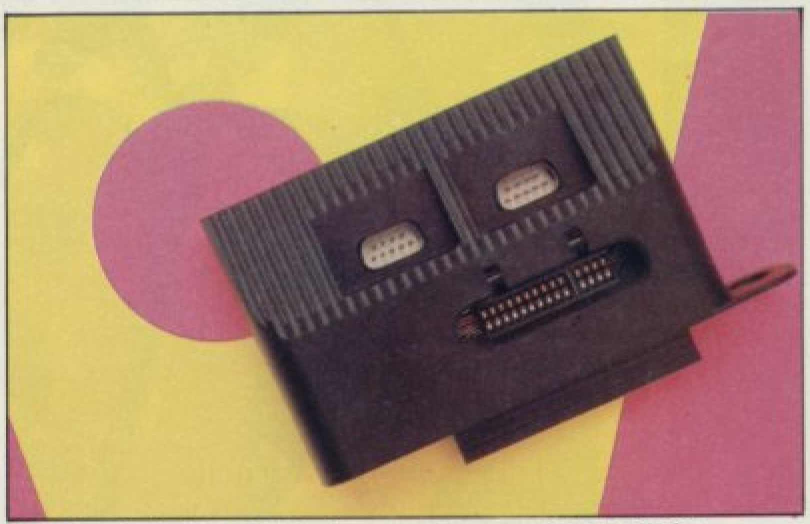
Ram Electronics, the Fleet-based hardware company, recently announced the launch of a new 'multi-purpose' Spectrum interface.

The device, modestly named the 'Ram Turbo', accepts Sinclair ROM cartridges and two joysticks of the standard 9-pin 'D' connector type. At this point you may be thinking this sounds rather like one of Sinclair's own Speccy add-ons — and you'd be right.

Although the Ram Turbo is £3 more expensive than the ZX Interface 2, it looks more futuristic and has a couple of extra design features. A unique built-in safety device prevents the user from inflicting expensive damage on the Speccy — the power cable can only be connected once the interface is in place. The Turbo also offers a full expansion bus (unlike Interface 2) which means there are no restrictions as to what other bits you may be thinking of bolting on the back.

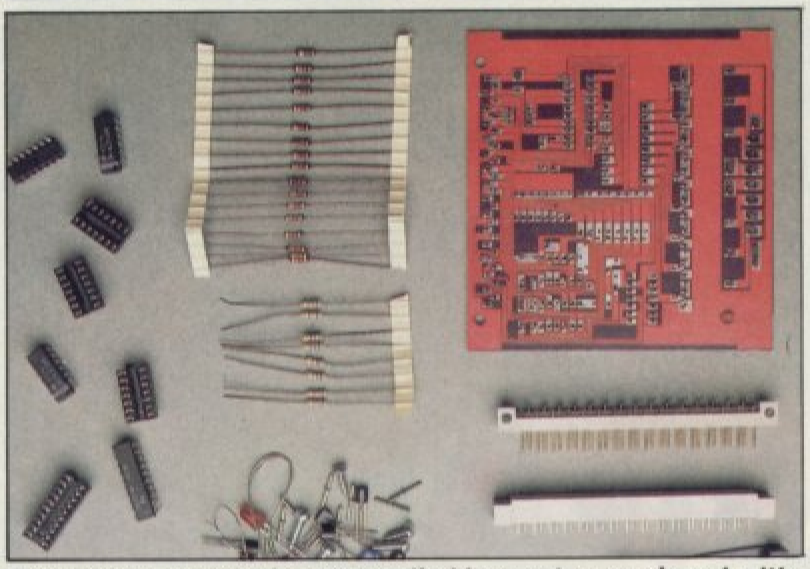
"We expect to sell over 50,000 Turbos in the first three months alone," says technical director Martin Shoebridge. "Technically it's miles ahead of any competitors (surely not a dig at Uncle Sir C!) but most people want it yesterday — not next month."

We trust this isn't just the latest contender for the Flying Pigs department. More information, etc., from Ram Electronics on 02514 5858.



The futuristic Ram Turbo incorporates a unique built-in safety device.

ON THE CARDS 1



Here are some of the items supplied for you to experiment with.

An alternative method to adding a multitude of cards to your Spectrum is the USP way, where one unit provides the following: the USP-PROT — which is a prototype device that allows the hobbyist to experiment with building boards; the USP-232D, a dual-channel RS232 serial interface; the USP-I/O, a general purpose parallel interface built around the Z80 PIO; and the USP-CENT, a kit for the I/O board to give you a Centronics interface.

USP can be contacted at U-Microcomputers Ltd, Winstanley Industrial Estate, Long Lane, Warrington, Cheshire WA2 8PR.

HOBBIT HACKERS

For all of you who bought *The Hobbit*, avidly read the book, tried to play the game and were *still* confused, Melbourne House has published a new book that could well answer your prayers (or may alternatively bring on a few nightmares).

A Guide To Playing The Hobbit splits up into three sections. The first offers the reader a broad outline of the general strategies and tactics involved, and the second and third sections give an increasing amount of guidance and detailed solutions to the problems encountered while playing the game. Melbourne reckons that the guide won't spoil the fun of it all because *the* solution is only offered among many other possibilities.

The Hobbit may have won the Golden Joystick Award for 'Strategy Game of the Year' in 1983, but it's really saying something when companies have to publish the Guide to the Game of the Book just so that people can play it properly!

A Guide To Playing The Hobbit is published by Melbourne House, written by David Elkan and costs £3.95; it should be available from bookshops now.

ON THE CARDS 2

New from Electronic & Computer Workshop is an interface system which allows you to slot in, via a motherboard, all these interface cards. Available now is: an output board, which will equip you with eight outputs for use with all kinds of switching applications; an A/D Convertor, that enables analog values to be read; an Optocoupler Input board, which makes possible the supervision of alarm units, detection systems, switches and time regulation on machines; a Centronics interface and a D/A Convertor that provides the opposite function to the A/D Convertor.

You can contact Electronic & Computer Workshop on 0245 262149.

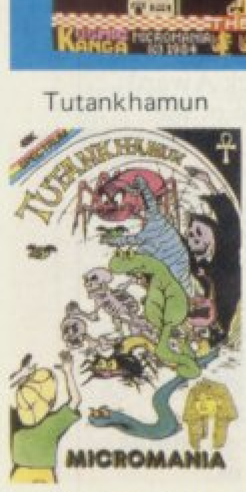




KOSMIC KANGA, a multi-screen, amazing, arcade standard original action game. Featuring superbly animated graphics and packed full of arcade features. Help KANGA find his space ship so he can return home to his planet. Leap about buildings, trees, clouds etc., pick up bonuses and throw boxing gloves at the enemies.

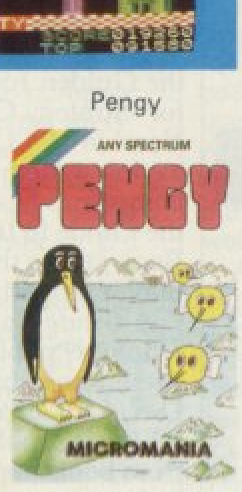
Keyboard or most joystick. **48K SPECTRUM** for most other Computers soon.

Tutankhamun



48K Spectrum

Pengy



16K/48K Spectrum

Invasion Force



48K Spectrum plus stack SLR

ONLY £5.95 EACH

Available from most good software retailers,
If your local software retailer is out of stock, send cheque/P.O. to Micromania, 14 Lower Hill Rd., Epsom, Surrey, KT19 8LT., and we will send your order by first class post free.

MICROMANIA

DEALERS: PHONE MICRODEALER UK 0727 34351 OR MICROMANIA (03727)20152



QUICKSILVA

...THE YEAR OF THE GAME LORDS.

NEW RELEASES!

COMMODORE 64
STING 64 £7.95
Author: **Anton Hinxman**
Hive-cave action!
Bertie Bee needs help defending the hive. Fight off the invading swarms, battle the bees and defend your Queen!

COMMODORE 64
BUGABOO (THE FLEA) £7.95
Author: **Indescomp**
Itchy action!
Jump your way out of the caves with Bugaboo the flea but beware of the fearsome Dragon as you jump around the exotic vegetation.

48K SPECTRUM
RAYMOND BRIGGS' THE SNOWMAN £6.95
Author: **David Shea**
An enchanting game based around episodes of Raymond Briggs' amazingly successful book.

48K SPECTRUM
ANT ATTACK £6.95
Author: **Sandy White**
Battle the Ants in the soft solid 3D city of Antescher.

48K SPECTRUM
DRAGONS BANE £6.95
Authors: **M. Preston, P. Hunt, R. Rose, D. Moore.**
A mythical graphic adventure in the dark and deadly halls of Earthstone Castle. Battle Dragons, Vampires, Sphinx, Zombies, Skeletons, Gryphons and other legendary beasts to rescue the beautiful Princess Paula.

48K SPECTRUM
FRED £6.95
Author; **Indescomp**
Action beneath the Pyramids!
Fearless Fred the Intrepid Archaeologist searches the creepy catacombs below the torrid tomb of 'Tootiecarmoon' for the terrible treasures amidst monstrous mummies, ghastly ghosts, bats and rats!

48K SPECTRUM • **TIME GATE** *Author: John Hollis* **£6.95**
COMMODORE 64 • **PURPLE TURTLES** *Authors: Mark & Richard Moore* **£7.95**
VIC 20 + 3K or 8K RAM • **SKYHAWK** *Authors: Steve Lee/Chartec* **£7.95**
DRAGON • **MINED OUT** *Authors: I & C Andrew/Incentive* **£5.95**
BBC MODEL • **THE GENERATORS** *Author: Dave Mendes* **£5.95**

All titles available from Quicksilva Mail Order P.O. Box 6, Wimborne Dorset BA21 7PY.

Quicksilva programs are available from: Boots, W.H. Smiths, J. Menzies, Microdealer, HMV, Hamleys, John Lewis, Computers for All and all reputable specialist computer stockists.

SEND S.A.E. FOR FULL COLOUR CATALOGUE

TURBO CHARGE YOUR SPECTRUM



Outperforms any Spectrum interface

The unique Turbo interface from Ram gives you all these features – and more – in one unit:

* A variety of interfaces including: Rom cartridges, two 9-way D plugs for standard joysticks, PLUS full expansion bus at rear.
* Compatible with Kempston and Protek protocols.
* Works with latest Quickshot Mk II auto rapid-fire joysticks!
* Choice of Rom cartridge or tape cassette software.
* Instant program loading with cartridge software.
* Built-in power safety device – unique to Ram Turbo.
* Full one year guarantee.
* Immediate availability – 24 Hr despatch on receipt of P.O./credit card details (cheques – seven days).
* Incredible value – only £22.95.

So don't wait around – simply complete the coupon and send it to us today.

Or call our credit card hot line on 02514 25252. (Access and Visa welcome).

Ram Electronics (Fleet) Ltd, 106 Fleet Road, Fleet, Hampshire GU13 8PA.

24 Hr despatch for credit cards and postal orders

Please send me:

_____ Spectrum Turbo Interface(s) at £22.95
+ £1 p+p (overseas orders £3 p+p)

_____ Quickshot II Joystick(s) at £9.95
(Only when purchased with Turbo – normally £12.95 + £1 p+p)

I enclose cheque/postal order or charge my Access/Visa for: £ _____

Name _____

Address _____

_____ Tel _____ Y.S.

To: Ram Electronics (Fleet) Ltd, 106 Fleet Road, Fleet, Hampshire GU13 8PA.

RAM TURBO

Trade and export enquiries welcome.

IT'S DIFFERENT FOR GIRLS

The powers that be at CCS have, in their ultimate wisdom, decided that girls don't *want* to play the same sort of games as boys; and based on this presumably unresearched piece of observation, the company's launched (as far as the feminist movement would be concerned) a highly dubious range of games for the fairer sex.

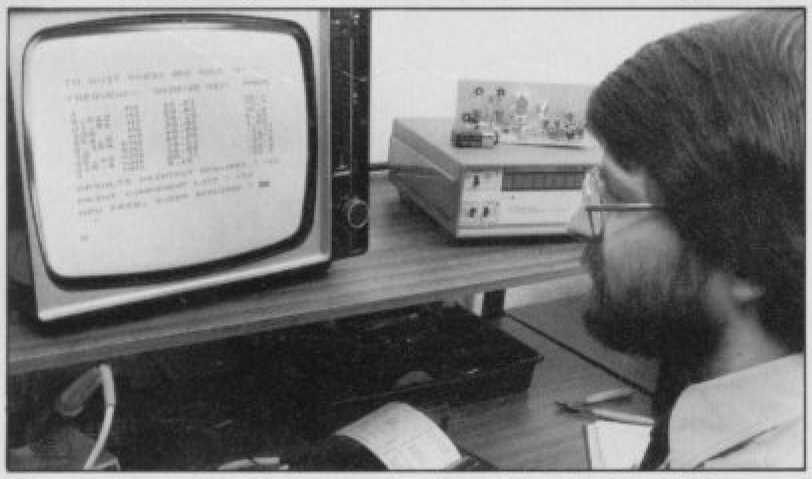
Hicksted/Mathsted is a showjumping simulation of that well-known event of the same name — but here the maths version involves getting over the jumps only when the answers given are correct; could this one be for all budding Princess Annes?

Jungle Adventure involves the player taking the part of a young jungle girl making her way home through the trees, while *Diamond Quest* is quite simply an adventure game full of colour and diamonds — which are, as we all know, a girl's best friend.

CCS describes the games as being less involved with killing monsters and more concerned with either bribing or avoiding them — traits which everyone knows to be thoroughly femine. Well . . . aren't they? Just to make sure we asked a fully paid up member of the female sex (a staffer) exactly what she thought of CCS's gender-orientated efforts. She said, "They were boring with no monsters; I didn't play any of them for long". She also admitted that what put her off most was the words on the packaging.

Anyway, if this seems to be your cup of idealogically unsound tea, *Hicksted* and *Jungle Adventure* retail at £6.00 each, *Diamond Quest* is £5.00. Further details from Case Computer Simulations Ltd, 14 Langton Way, Blackheath London SE3 7TL.

WOT A CAD!



This program should bring professional electronics CAD facilities within everyone's reach.

Now that the 'Computers in Schools' scheme is really under way, Number One Systems has announced the launch of a Spectrum version of its AC Linear Circuit Analysis Program.

Number One says the version for the Spectrum came about particularly as a result of requests from undergraduate students and school Science teachers. It claims the program brings professional electronics CAD facilities within the reach of us all.

Whether a program of such specific use will manage to sell enough to justify mass production is a difficult question to answer, but nonetheless, Number One seems to believe in it; have a natter with the people there on 0480 61778.

SINCLAIRWATCH

Well, the latest deadline for the first QL deliveries has come — and gone — and still (at the time of writing) no sign of the machine itself. After the imaginative 'End of February', came the 'by the end of March' announcement, and Sinclair Research stuck to its story up to the last possible moment — and then admitted it couldn't keep to it (surprise, surprise). The latest word is that customers should get QL-ed by the dates specified in individual letters, estimates which currently range from the end of May through to the end of July (although potentially more worrying is the fact that the year of delivery has been omitted!). An utter sceptic was recently heard misquoting the famous line: "There are three kinds of lies — lies, all lies and Sinclair Research delivery dates".

The exact reasons for the ever-lengthening delays are difficult to confirm. All those in the know are keeping very quiet — well, almost all — apparently someone in Ferranti recently discovered a 'major hardware fault' in the QL's design that had escaped earlier detection. Other rumours claim that the operating system, QDOS, won't fit into the 16K ROM space allocated — and that major routines like multi-tasking and windows are going to have to be left out. This may be not come as much of a surprise to some, particularly as Apple has reported difficulty fitting the Mackintosh operating system into a 64K ROM.

When (being positive) the QL is manufactured in quantity, one thing is sure — just like the Spectrum it's going to take time to clear the backlog of orders. Therefore, the chances of it being available through retail outlets before the New Year look rather remote. Naturally, though, the de-luxe double-page full-colour adverts continue to be booked, which in turn lengthens the queues, which in turn lengthens the potential delay, which in turn increases the interest accruing on the customer's hard-earned lolly. To compensate for this loss of interest, customers are to be offered a free and desirable gift of some sort, the nature of which is still to be decided. By the look of things, Sinclair Research may have plenty of time in hand to make the decision. The Spectrum freebie (to make up for its delivery delay) was a £10 voucher towards a ZX Printer or paper — I don't somehow think they could use that again!

Actually examining the infamous QL adverts with a bit of honest hindsight reveals some interesting quotes. For example, "nothing like it exists anywhere" — which is true enough. Also, it suggests that you "Get yourself a QL at the earliest possible moment" — which is currently likely to be August (at least according to the Gospel of Sir Clive). And then, of course, there's the evergreen "delivery may take more than 28 days".

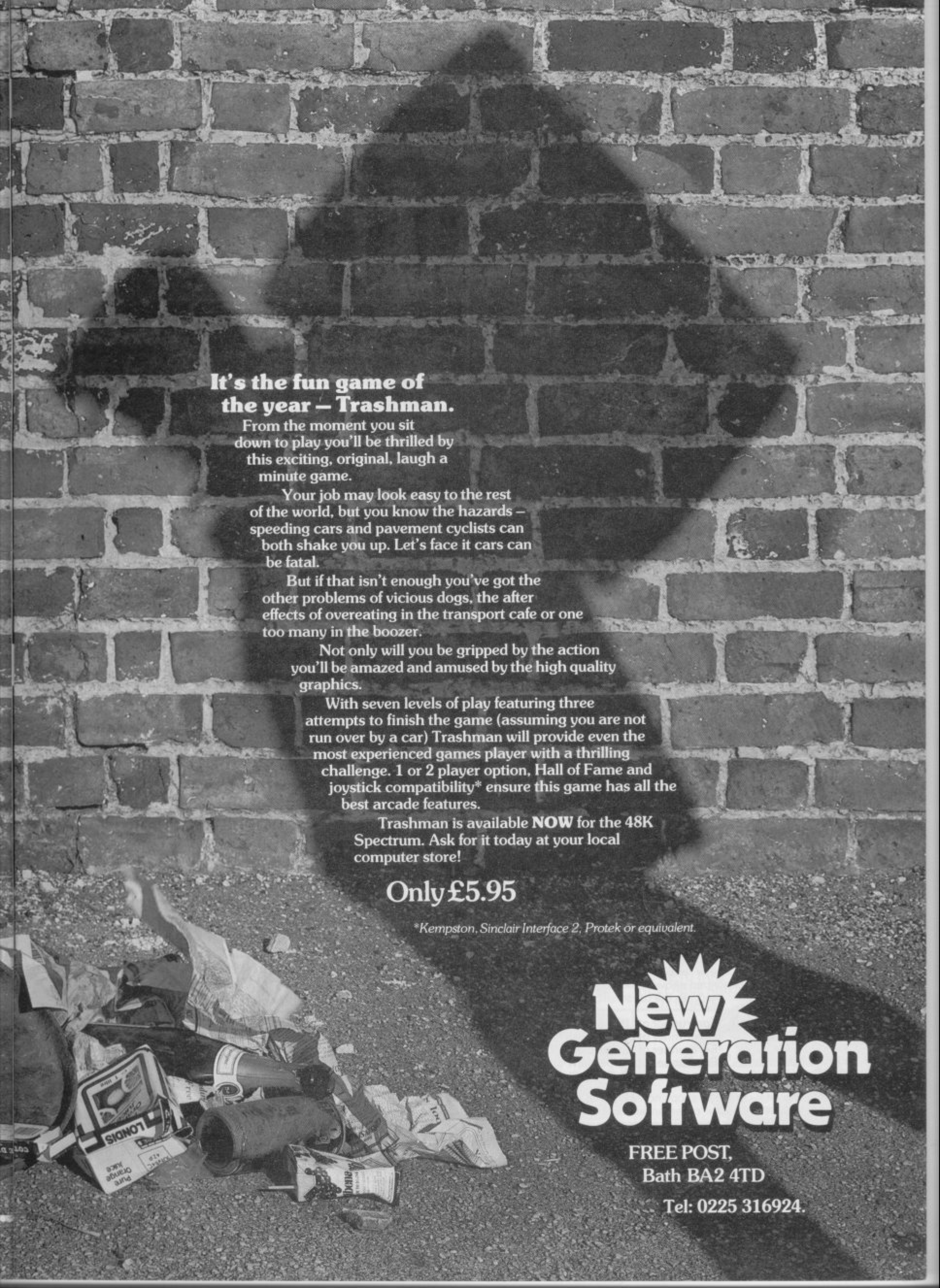
Also mentioned in the ads is the bit about extending your credit limit if you order by credit card, and the reason for this is now clear. The rules of the card companies insist that money can only be obtained when despatch of goods is imminent. Therefore, if Oswald the Optimist orders a QL by Access in, say, April, when he's fortunate enough to have the odd £400 hanging around his account, by the time his QL is actually despatched, he could well have blown the dough. Thus, when Sinclair Research checks his status, Access will normally refuse the order and back goes Oswald to the rear end of the queue and a sale has bitten the dust. But . . . under Sir Clive's new wonder scheme, Access will probably allow you to exceed your limit by the price of a QL. Clever innit?!

DREAMY SCREENS

If you consider yourself a 'serious' Spectrum user and you're into displaying more than 24 rows of 32 characters at any one time then, for a mere £5.95, Timedata can supply you with a new software product called the *HI-T Spectrum Screen Enhancer*. This program gives you up to 32 lines of 64 characters — a total of 2048 characters on the screen; 'form-filling' where user input can be to any position on the screen without disturbing any other displayed data; print windows for confining any printing and scrolling to any rectangular area on the screen; offset printing to move any characters up by one to five pixels for superscripts, subscripts, and so on, and re-definable print comma spacing for flexible tabulation.

HI-T is a fully relocatable machine code program which can be used on either 16K or 48K Speccys (with or without Microdrives) and is available for £5.95 from Timedata Ltd, 16 Hemmells, High Road, Laindon, Basildon, Essex SS15 6ED.


Trashman
GSH
Guild of Software Houses
New Generation products are sold according to their terms of trade and conditions of sale.



It's the fun game of the year – Trashman.
From the moment you sit down to play you'll be thrilled by this exciting, original, laugh a minute game.
Your job may look easy to the rest of the world, but you know the hazards – speeding cars and pavement cyclists can both shake you up. Let's face it cars can be fatal.
But if that isn't enough you've got the other problems of vicious dogs, the after effects of overeating in the transport cafe or one too many in the boozer.
Not only will you be gripped by the action you'll be amazed and amused by the high quality graphics.
With seven levels of play featuring three attempts to finish the game (assuming you are not run over by a car) Trashman will provide even the most experienced games player with a thrilling challenge. 1 or 2 player option, Hall of Fame and joystick compatibility* ensure this game has all the best arcade features.
Trashman is available NOW for the 48K Spectrum. Ask for it today at your local computer store!
Only £5.95
*Kempston, Sinclair Interface 2, Protek or equivalent.
New Generation Software
FREE POST,
Bath BA2 4TD
Tel: 0225 316924.

THE YS TOP 20

No more chart fantasy — here it is, the first popularity listing to be based entirely on the loves, loathings and prejudices of *Your Spectrum* readers.

As promised, our hopelessly inadequate prize of three current software releases goes to the first name out of the bag — which this month is Carl Tanner of Portsmouth in Hants. Thanks Carl, and everyone else who took the trouble to send in the polling slip below. KEEP 'EM COMING!

You'll notice there's an extra section on the slip — that's right, we've introduced a second 'chartette' entitled Top Turkeys. No guesses what that's all about, just tell us your three all-time dog games and the executioner will do the rest.

YS TOP 20 READER POLL

My top five raves on the Speccy are:

1 __________

2 __________

3 __________

4 __________

5 __________

My top three dogs on the Speccy are:

1 __________

2 __________

3 __________

I understand that the Surgeon General has deemed these charts to be hazardous to my mental condition.

Post with haste to:

YOUR SPECTRUM CHARTS
14 RATHBONE PLACE
LONDON W1P 1DE

YOUR SPECTRUM

THIS MONTH		LAST MONTH		
1	▲	—	**JET SET WILLY** SOFTWARE PROJECTS	SEQUEL TO MANIC MINER. YOU CAN JOIN WILLY IN 60 MORE EXCITING LOCATIONS.
2	◀▶	2	**MANIC MINER** SOFTWARE PROJECTS	JOIN MINER WILLY IN HIS QUEST FOR THE MISSING KEYS, IN 20 BIZARRE AND COLOURFUL CAVERNS.
3	▲	5	**DEATHCHASE** MICROMEGA	REMEMBER THE 'BIKE CHASE' SCENE FROM 'RETURN OF THE JEDI'? ALL HAS BEEN RECREATED.
4	▲	—	**ANT ATTACK** QUICKSILVA	AN AMAZING 3D ADVENTURE IN THE CITY OF ANTESCHER THAT HAS YOU RESCUING YOUR DUMB MATE FROM THE KILLER ANTS.
5	▲	11	**FIGHTER PILOT** DIGITAL SYSTEMS	A 3D FLIGHT SIMULATION PROGRAM THAT SITS YOU BEHIND THE CONTROLS OF A FIGHTER PLANE.
6	▼	3	**ATIC ATAC** ULTIMATE	A FULL-GRAPHICAL ADVENTURE IN AN ARCADE FORMAT, BUT WATCH OUT FOR THE WANDERING GHOSTS!
7	▲	13	**JET PAC** ULTIMATE	FIRST YOU BUILD A SPACESHIP, THEN YOU GO 'STAR-TREKKING' AROUND, COLLECTING GEMS FROM OTHER WORLDS.
8	▲	—	**SCUBA DIVE** DURRELL	PUT ON YOUR WET-SUIT AND PLUNGE INTO THE ICY DEPTHS IN SEARCH OF PEARLS AND TREASURE.
9	▼	6	**CHEQUERED FLAG** PSION	JOIN THE GRAND PRIX CIRCUIT AND RACE (SMASH UP?) SOME OF THE GREATEST TRACKS IN THE WORLD.
10	▲	—	**CHUCKIE EGG** A&F SOFTWARE	FARMER GILES HAS TO ROUND UP ALL THE EGGS BEFORE THE NASTIES GET OUT AND EAT UP ALL THECORN. BEWARE OF THE CRAZY DUCK!
11	▲	12	**LUNAR JETMAN** ULTIMATE	A DEMANDING ARCADE/ ADVENTURE THAT TAKES PLACE ON THE FROZEN WASTES OF THE MOON.
12	▼	4	**STONKERS** IMAGINE	A COMPLEX GRAPHICAL WAR SIMULATION WHERE THE TASK IS TO TAKE OVER THE WORLD — STRICTLY FOR MEGALOMANIACS
13	▼	1	**HUNCHBACK** OCEAN	A FAST MOVING ARCADE CONVERSION THAT HAS YOU JUMPING THE WALLS OF THE CASTLE TO SAVE ESMERELDA.
14	▲	15	**PYRAMID** FANTASY	THERE ARE 120 COLOURFUL ROOMS AND ALL YOU HAVE TO DO IS COLLECT THE CRYSTALS — THE TROUBLE IS IT'S RAINING ALIENS.
15	▲	—	**TRASHMAN** NEW GENERATION SOFTWARE	TRY AND EMPTY THE DUSTBINS IN THE FASTEST POSSIBLE TIME, AVOIDING SNAPPING DOGS AND KILLER BICYCLES AND DON'T DRINK TOO MUCH IN THE PUB.
16	▲	—	**FOOTBALL MANAGER** ADDICTIVE GAMES	A SOCCER SIMULATION WHERE YOU HAVE TO TAKE YOUR TEAM THROUGH TO THE LEAGUE AND FA CUP FINALS.
17	▲	—	**TRANZ AM** ULTIMATE	DRIVE YOUR TRANZ AM ALL THE WAY ACROSS AMERICA. PICKING UP CUPS AS YOU GO AND AVOIDING BLACK CARS.
18	▼	9	**MR WIMPY** OCEAN	A CHANCE TO DISCOVER THE HORRID TRUTH OF HOW BURGERS ARE MADE — AN ARCADE CONVERSION.
19	▲	—	**THE HOBBIT** MELBOURNE HOUSE	AN ADVENTURE BASED ON THE BOOK BY TOLKIEN. YOU TAKE THE PART OF BILBO BAGGINS IN SEARCH OF DRAGON'S TREASURE.
20	▼	10	**ALCHEMIST** IMAGINE	A WIZARD PROGRAM WHICH MANAGES TO BLUR THE BORDERS BETWEEN ADVENTURE AND ARCADE.

FORUM

If you've got something you want to tell the world about then write to Forum, Your Spectrum, 14 Rathbone Place, London W1P 1DE.

SOFT ROM CORRECTIONS

Concerning your article on the Soft ROM by Mike Lord. Having built the project, I discovered that 64 x 1 (4164) DRAMS would not work in the circuit as given. The problem arises from the Z80 refresh system which only puts out a 7-bit refresh address (A0 to A6); 4164s require an 8-bit refresh (that is, a 256 refresh cycle).

My solution was to ensure that address lines A0 to A6 are strobed by RAS and A7 to A13 by CAS. A14 and A15 are 'don't care'.

Unfortunately, anyone wishing to use this project to upgrade to 48K will not be able to do so without changing the refresh system.

Also, when using the Soft ROM with a 48K Spectrum, I would recommend an external 5v supply using the 9v line from the edge connector to drive it.

I've designed a double-sided PCB for this project; any readers interested should write to the address below.

P Giblin, 99 Staley Hall Road, Stalybridge, Cheshire SK15 3DP

INCOMPATIBLE LPRINT

Although I find your magazine generally very good, one of your articles contains an inaccuracy which has caused me a lot of problems.

In the article 'Getting into Print' you state that the ZX LPRINT works with the ZX Microdrive, which is not true at all. I have both these items and they will not work together, or rather the ZX LPRINT works, but the Microdrive won't while the ZX LPRINT is connected.

I contacted Euro-electronics (makers of the ZX LPRINT) and after much evasive talk the company finally conceded that the product is incompatible with the Interface 1 unit. They do offer an adaptor at £4.50 which allows the ZX LPRINT to be switched out when the Microdrive is in operation; but will this allow such things as CAT#3;1 — I doubt it!

David Leckie, Fort William

ANT ATTACKS

I'm writing to you about this fantastic competition which you held, together with Prism. When I read the rules I hastily got out my copy of (original) *Ant Attack* and proceeded to battle my way through the different levels, finally reaching the magic tenth level. Great, I thought, only another three times and I've cracked it. But wait what's this, on the second time she was in the same place. So off we went again, the intrepid rescuer in search of his damsel, and eventually after about twenty attempts I finally found all four places (The Forum, The Oxymine, Artant's Villa and Droxtrap, by the way). Now all I had to do was wait until 30th March and answer the next part.

Finally 30th March arrived and I quickly grabbed *The Daily Telegraph* off the doormat and hastily scanned the classified ads. Alas I couldn't find it, so I browsed through all the pages as the time slowly ticked away. But still I couldn't find it so I rushed out and bought a copy of *The Guardian* and there on page 25 sat the ad. I quickly read it and loaded up my copy of (original) *Ant Attack* yet again, and searched high and low for the mysterious object, after about two hours I found the box marked 'AMMO'.

Now, I thought, I just have to wait until noon to ring through my answers. But oh no!! — Telecom had different ideas. Every time I rang the number given I got the engaged tone. After 12.45 I rang the operator: "all lines to London are engaged please try later" was the reply — I could have screamed. All my hard work to waste and still at 13.45 the line was engaged. Time to give up, I thought.

So that was when I decided to write this letter. Don't you think it's a bit unfair for the people who live outside London, giving us no chance of getting through. Maybe a later time would have been a little better, you know! Maybe when no one is at work and all the lines are not busy. Somebody from the South probably won the competition as us poor Northeners could not get a reply.

Next time, if there is one, give everyone a fighting chance.

I now go to ceremoniously burn my copy of *Ant Attack* kindling it with the second copy of *YS* (I have to take it out on something and it was, after all, the root of all my frustration).

Andrew Hatton, Macclesfield

First, our thanks to The Daily Telegraph *for not publishing the ad in the Northern Editions of the paper — nice one chaps! Actually the first call to come in wasn't from London. However, a wrong answer did give it to the second caller — from London. Apologies to all those who felt aggrieved at hearing only the engaged tone. With (according to BT) something like 10,000 callers trying to get through, it was inevitable that 9,999 wouldn't be able to make it at any one time. The odds for winning with this kind of telephone tombola must, of course, be the same as postcards out of a hat and we hope the approach gave the competition a refreshing twist. Ed.*

PARDON THE PLUG

As the author of the first book about the QL, *The Sinclair QL Companion,* I must point out that Andrew Pennell is wrong to compare SuperBasic with Pascal. SuperBasic has little in common with Pascal, but shares a great deal with Algol 68. All this is explained in more detail in the *QL Companion.*

In the *QL Companion* I refer to Seymour Papert (author of Logo) because his ideas on programming and computers are relevant to any discussion of structured programming. I have always felt that computers and programs should be kept away from computer studies departments, and Sandy Dewhurst's philistine review of computer education texts reinforces that view. When one reads on the one hand he cannot cope with Papert, and then finds that "The Spectrum is quite a unique machine . . .", all is explained.

Boris Allen, Bramhall

BETTER SOUND & VISION

I've discovered a simple way of modifying the Spectrum to output a signal suitable for the Video In socket of a VCR or TV. The effect on picture quality is startling and it also enables you to run Spectrum sound through to your TV.

Monitor Conversion

1. Remove the five screws from the base of the Spectrum.
2. *Carefully* lift off the keyboard and leave it resting on the front half of the PCB. You must be gentle to avoid damage to the ribbon connectors.
3. You'll see a silver box on the top left-hand corner of the circuit board. This is the modulator (it has the word 'Aztec' written on it). Two wires go into the left side of the modulator. You'll be soldering the inner core of a length of coaxial cable to the one nearest the rear of the computer (the one which disappears into a polythene insulator).
4. Strip 1½ inches of the outer insulation from the coaxial cable and twist back the surrounding wire. Cut the twisted wire into a ½-inch length, dip it in flux and tin with solder. Strip back just under ¼-inch of the insulation from the inner core, dip it in flux and tin it.
5. Solder the inner core of the cable to the wire which enters the modulator nearest to the rear of the computer.
6. Solder the outer sheath to the outside of the aerial socket.
7. Lead the cable out of the hole provided for the TV socket and carefully replace the screws.
8. Put a BNC plug on the other end of the cable and, hey presto, you now have a composite video output on your Speccy.

Sound Conversion

1. Remove the jack plug in the computer MIC socket. You'll have to cut it off because it's a sealed unit.
2. Solder the original wire and another wire to a 3.5mm jack plug. This will leave you

with two wires coming out of the plug.
3. Attach a suitable plug to the loose wire. Its type will depend on the sort of sound input socket on your VCR/TV.
4. Push the new plug into the Sound In socket on your equipment. When you select Video the sound should come through the TV/monitor speaker.

I hope your readers will find this interesting. All the best.
NE Salt, Crayford

A PEEK BEHIND

Having just read the third issue of *Your Spectrum*, I felt compelled to write regarding the masterpiece of journalistic misprint on page 29, entitled 'A PEEK in Time . . .'.

First of all, the traditional method of PEEKing 2-byte system variables appears to have been extended in your opening paragraph:

```
PEEK n+1*256*PEEK(n+1)
```

One multiplied by 256, anyone? A minor point perhaps, but since the whole point of the exercise is to save time and use, 'two less bytes of precious memory' (not a major consideration, I'd have thought, even on a 16K Spectrum) but perhaps one still worth mentioning.

This little gem is immediately followed up by the real killer:

```
PEEK n+1*256+PEEK n
```

Let me see now . . . by my reckoning, that reduces to:

```
2*PEEK n + 256
```

which is not the same thing at all!

Turning now to that listing with the wonderfully spaced line numbers, the first routine translates to the general form:

```
PEEK (n+1)*256 + PEEK n
```

. . . and the second to:

```
PEEK n + 256 * PEEK (n+1)
```

Oh look Daddy! The clever man has written the line backwards. That's bound to make it faster, isn't it?
Captain Critical, Dartford

OK, humble pie all round (again). Mr Mada's theories are obviously slightly off-track. But, as Ian points out at the end of that piece, the only reason the routine is slightly faster is because the brackets have been removed, not because it was written backwards. Anyway, consider yourself given a Silver Train-Spotter award. Troubleshooting Pete.

STANDARD CONFUSION

There seems to be some confusion in your article on the Spectrum RS232 Interface (Page 29, *YS* issue three). The problem must have occurred because of the non-standard labels that Sinclair Research has used. Your reviewer claims that the interface is non-standard, but this is not the case — he's just wired the RS232 back to front (I made the same mistake myself). The Interface 1 manual states the following connections for the RS232.

Pin number	Sinclair name	Normal RS232 name
2	TX	RXD
3	RX	TXD
4	DTR	CTS
5	CTS	DTR
7	GROUND	GROUND
9	+9v	+9v

As you can see from the table Sinclair Research has named its connections according to the place they go, not what they are. These unusual names have confused everyone I've met who has tried to use the RS232.

I've had a Speccy talking to a CP/M type computer for a couple of months now and communication is bi-directional at 9600 baud. The only problem is that the expensive machine has a non-standard handshaking arrangement, which forced me to make up my own lead rather than be able to use Sinclair Research's.
Jon Ritman, London N13

Hmm, the Bear Bovver *person, if I'm not very much mistaken. Thanks a lot, Jon, for pointing out the error of our ways. Ed.*

PRINTING PROBLEMS

I must first congratulate you on the production of an excellent magazine devoted entirely to the Spectrum. I welcome such a magazine as other Sinclair mags tend to be aimed at a slightly younger readership who wish to spend time keying in programs and nothing else. However, I digress.

My main reason for writing this letter to you was to highlight a point arising from reading your excellent Issue 3.

There was an article on page 9 about the new Kempston printer interface in which Peter Shaw said that apart from the Fuller Printerface, the Kempston was the only one with software built-in. Not so!!! Euroelectronics produce an interface LPrint III which not only has all its operating software built-in but includes an RS232 as well as a Centronics interface as part of the package. Even better for those of us who, after buying a dot matrix printer are a little short of those green things, is the fact that it costs only £36.00. Admittedly that does not include a cable but for those who are not able to make one up they will provide one for a charge which still does not bring it up to the cost of the Kempston.
Steve Brokenshire, Bridgend

Sorry Steve, when writing that piece on the Kemmy interface, the LPrint III had not been brought to my attention. Point taken. You know, I'm rapidly going off this humble pie. Troubleshooting Pete.



NURD OF THE MONTH

Please, I am interested in your magazine *Your Spectrum*.

Please forward the magazine *Your Spectrum* for 1983 numbers 2, 3, 5, 6, 8, 10, 11 and 12.
L Vilfan, Bled, Yugoslavia

We'd love to send you the missing copies from 1983 — trouble is, we weren't around then. By the way, what's wrong with issues 1, 4, 7 and 9? Ed.

GILDING THE LILY

I would be grateful if you could supply me with the following information. Are there any software packages for the ZX Spectrum which enable the user to produce screen displays of 42, 46 or 51 characters in each row and, if so, where would it be possible to purchase such a package? I really enjoy your magazine and I am glad to hear that you have gone monthly.
Dermot Connelly, Co. Londonderry

Timedata have recently released a utility called HI-T which gives various screen formats for the Speccy. See Frontlines for more details. Troubleshooting Pete.

IMPORTANT INPUT PORTS

I have recently seen a number of comments in the Spectrum orientated press about the values returned from the keyboard input ports. That the 'unpressed' value in some machines is 255 and in others is 191. I decided to investigate my own machine which is a Series 3 model, so my findings may not apply to other Spectrums.

Rather than just use PRINT IN 32766 (or whatever) I wrote a little program which initialised the screen and then, from within as fast a loop as I could devise, continually updated the value read from all eight keyboard ports as well as the curent PEEK 23560 value and INKEY$.

The results were not what I expected. Sometimes the 'unpressed' value from all ports was 255, sometimes 191; and sometimes the values flickered between 255 and 191. The variation (of 64) has to be due to Bit 6, which (if I read my Spectrum book correctly) represents the EAR socket. Is the CPU continually testing it? The flicker seems to be the same as the red/cyan border change that you see when about to LOAD and the tape has not been started.
Mike Minchin, London W11

This is a problem which a lot of software houses kicked up a fuss about when Sinclair released the issue 3 machines last year. The problem cannot be cured however, unless anybody out there has some suggestions? Troubleshooting Pete.

```
   1 REM SAVE "Peek Input" LINE 10
  10 CLS : PRINT AT 0,0;"INKEY$=";AT 3,0;"Peek 23560=";AT 7,0;"1 2 3 4 5";AT 8,0;"IN 63486=";AT 7,16;"6 7 8 9 0";AT 8,16;"IN 61438=";AT 10,0;"q w e r t";AT 11,0;"IN 64510=";AT 10,16;"y u i o p";AT 11,16;"IN 57342=";AT 13,0;"a s d f g";AT 14,0;"IN 65022=";AT 13,16;"h j k l E";AT 14,16;"IN 49150=";AT 16,0;"C z x c v";AT 17,0;"IN 65278=";AT 16,16;"b n m S █";AT 17,16;"IN 32766="
  20 LET A$=INKEY$: IF CODE A$=13 THEN LET A$="(line return)"
  30 IF A$="" THEN LET A$="█"
  40 PRINT AT 0,7;A$,,AT 3,11;PEEK 23560,,AT 8,9;IN 63486;AT 8,25;IN 61438;AT 11,9;IN 64510;AT 11,25;IN 57342;AT 14,9;IN 65022;AT 14,25;IN 49150;AT 17,9;IN 65278;AT 17,25;IN 32766: GO TO 20
```

UPGRADE YOUR ZX SPECTRUM NOW!

The "CHEETAH" 32K Rampack simply plugs into the user port at the rear of your computer and increases the memory instantly to 48K.

- Fully compatible with all accessories via rear edge connector.
- No need to open computer and invalidate guarantee.
- Why send your computer away and wait weeks for upgrade.
- Fully cased, tested and guaranteed.

Why wait any longer?

ONLY **£39.95** inc. VAT and p&p



NOW MAKE YOUR SPECTRUM AND ZX81 TALK!

The CHEETAH "Sweet Talker" just plugs into the back of the computer using the existing power supply. Based on an allophone system you can easily program any word, sentence or phrase. Fully cased, tested, guaranteed and compatible with all accessories via rear edge connectors. Complete with demonstration cassette and full instructions. No more lonely nights!

Simply incredible at **£29.75**

Also available:
16K Rampack for ZX81 ...... **£19.75** 64K Rampack for ZX81 ...... **£44.75**

Prices include VAT, Postage and Packing. Delivery normally 14 days. Export orders at no extra cost. Dealer enquiries welcome.

Send cheque/PO now to:-

CHEETAH MARKETING LIMITED

Dept YS1, 24 Ray Street, London EC1 Tel: 01-833 4909

Please quote when ordering whether **Spectrum** or **ZX** owner.
32K Rampack and Sweet Talker available from larger branches of John Menzies Boots *and* WHSMITH

Computers for All dealers.

TT-S: SPECTRUM TOOLKIT £7.95
For the 16 or 48K Spectrum. 5 programmer's programs on one cassette.

★ **GAMMA:** A relocatable, microdrive compatible BASIC extension giving improved editing plus 10 new keywords including intelligent RENUMBER and MOVE as well as DELETE, FIND, TRACE, FREE, AUTO and VARIABLES.
★ **SCREEN:** A High resolution drawing utility.
★ **UDG:** A powerful aid to creating user-defined graphics characters.
★ **TAPE:** Reads valuable information from tape file headers.
★ **RAM:** A thorough memory test program.
(A Dutch version of TT-S is available from AMB Software)

HI-T: SPECTRUM SCREEN ENHANCER £5.95
Gives the option of using 32 rows of 64 characters each on the screen of a 16 or 48K Spectrum, with text 'windows', superscripts & subscripts and INPUT AT any part of the screen. A fully relocatable microdrive compatible machine code program, integrated with the Spectrum's normal PRINT, INPUT and LIST commands. Invaluable for serious Spectrum users.

BOOKS: Exploring Spectrum Basic: ISBN 907892 03 5: £4.95
Explorers Guide to the ZX81: ISBN 907892 02 7: £4.95

ZXS SPEECH SYNTHESISER for the Spectrum or ZX81: £24.99
ZXM 3-CHANNEL SOUND BOX for the Spectrum or ZX81: £29.95

★ ★ ORDERS DESPATCHED WITHIN 48 HOURS OF RECEIPT ★ ★

Prices inclusive of VAT and U.K. P&P. Overseas customers add £1.50 per item for surface mail. Access/Visa accepted.

TIMEDATA Ltd., Dept N
16 Hemmells, Laindon, Essex SS15 6ED
Tel: (0268) 418121

HIRE SPECTRUM SOFTWARE

CHOOSE FROM OUR VAST RANGE
FREE 26 page CATALOGUE with full details of our range of over 280 different programs for the ZX SPECTRUM. All hired with the consent of the publishers. Whether you prefer to ZAP aliens, get lost in an ADVENTURE game, EDUCATE your children, CALCULATE the family finances or COMPILE machine code programs WE HAVE THE RIGHT TAPES FOR YOU TO HIRE.

FAST SERVICE
We stock up to 60 manufacturers' original copies of each tape (over 3000 tapes in stock), and can normally despatch your tapes the day we receive your order, and we always use first class post.

LOWEST PRICES
We believe our prices are the lowest of any library, yet we offer the LARGEST range of titles. Hire up to 3 tapes at a time, for 2 weeks, for only £1.07 each (inc. VAT & p&p).

FREE TAPE HIRE
For a limited period, we will send your first tape (your choice) FREE, so join our library now. You can't afford NOT to! If you've tried another library and are wary of our claims send a stamp and we'll gladly send you a copy of our catalogue - we're confident you'll join when you see it!

NATIONAL SOFTWARE LIBRARY
200 Mulgrave Road,
Cheam, Surrey SM2 6JT

I enclose cheque for £6.00 for LIFE MEMBERSHIP and I understand that my first tape hire will be FREE. If, within 28 days, I'm not delighted with your service you'll refund my membership fee.

Name ______________________

Address ______________________

YS4

GEM SOFTWARE

OH MUMMY!!
Your party of archeologists enter the pyramid in search of the mummies of the pharoah's household. There are five burial chambers containing treasure, secret scrolls, keys, royal mummies and some nasty surprises. To pass from one chamber to the next, you must find the key and royal mummy, but beware, you may uncover a guardian, who will chase your group to avenge your desecration!!
A Machine Code game for those with nerves of steel and great courage.

Price (ZX81 & SPECTRUM) £4.95

DISCO DAN
Can poor Dan decontaminate the atomic fuel rods in time?! In this new 3D machine code game, you control Dan Dan The Atomiser Man, jumping him from disc to disc, trying to reduce the ever increasing radiation levels, in the atomic pile. However, Up'n'Atom'Arry and his gang of vagrant particles are thirsting for the energy in your back pack, which makes life a bit awkward at times. Then there's Ron the Rapacious Robot, not the sort to meet in a dark alley (or anywhere else for that matter)!!

Price (48K SPECTRUM) £4.95

All prices include postage and packing. All cassettes despatched within 48 hours of receipt of order.
Access orders welcome.
DEALER ENQUIRIES WELCOME

GEM SOFTWARE, UNIT D, THE MALTINGS, STATION ROAD, SAWBRIDGEWORTH, HERTS.
Telephone (0279) 723567/723518.

THROUGH THE SQUARE WINDOW

Lose yourself in machine code with the maestro herself.* Toni Baker *takes you one step nearer creating the perfect arcade game with another riotous routine.

This month I'm presenting an interesting little routine that fits happily into many a video game. The gist of it is really very simple — a 'window' appears in the top right-hand corner of the screen and on it you can see part of a maze. The cursor keys: '5', '6', '7' and '8' — without shift — will scroll the window over the maze in all four directions (actually eight directions because you can go diagonally by pressing two keys at once). In other words, the window allows you to see a smaller part of a much larger maze — by moving the window you can cover the whole maze, but you can never see all of it at once.

So, what can you do with it? Well, as it stands you can probably have a bit of fun trying to find your way out of the maze — you start at the 'S' character and finish at the 'F'. The fact that you can only ever see part of the maze doesn't spoil the game — it just makes it that much harder! But if you wanted to, you could adapt the routine (or add more Basic) and make quite a comprehensive game. The window is always printed in the same place in black INK on white PAPER, but note that the normal PRINT position (as well as PAPER and INK colours) is totally unaffected by the machine code.

The Basic is completely separate from the machine code. By calling the machine code from Basic you will scroll the window either not at all or by just one square; thus, if you want it to move continuously, all you have to do is repeatedly call the machine code routine, over and over again. Of course, any number of Basic lines may be placed between these calls.

At present the size of the maze is 32 squares along each side. The program assumes that the maze is always a square, but it requires only minor alterations to produce a different size of square — and, of course, if you use different data to the stuff I've made up, then you can create a different maze.

SERIOUSLY NOW

Just on the off chance that there are any serious machine code programmers reading all this, you may be interested in the 'CALL HL=HL*DE' instruction used in the program. This is a subroutine in the ROM which, as its name implies, multiplies HL by DE, leaving the result in HL. The BC and DE registers are unaffected, but the A register is corrupted.

Once again, I've put question marks in the listing of the Hex codes, instead of absolute addresses — so you can locate the machine code anywhere you want to. Once you've decided where to put it you have to replace the question marks by the address of the instruction which has the *label* that I've specified. In other words — where it says 'CD???? CALL INC_COORD', you must replace the question marks with the address of the instruction labelled 'INC_COORD'. This address must be in Hex and with the bytes in reverse order.

There are two variables used by the program which you can POKE from Basic if you want to. These are X_COORD and Y_COORD which store respectively the X and Y co-ordinate of the top left-hand corner of the window (relative to the top left-hand corner of the maze). X_COORD is stored in address 23728, and Y_COORD in 23729.

So there you have it! No longer need you spend tiresome hours running round Hampton Court — now you can spend those tiresome hours wearing out your fingers instead. Happy headscratching!

CODE		ASSEMBLER	COMMENTS
7E	INC_COORD	LD A, (HL)	A:= co-ordinate to change.
FE18		CP Length-8	
C8		RET Z	Return if co-ordinate is at the maximum.
34		INC (HL)	Increment co-ordinate.
C9		RET	
7E	DEC_COORD	LD A, (HL)	A:= co-ordinate to change.
A7		AND A	
C8		RET Z	Return if co-ordinate is at the minimum (zero).
35		DEC (HL)	Decrement co-ordinate.
C9		RET	
3EF7	START	LD A, (F7)	
DBFE		IN A, (FE)	Scan segment three of the keyboard.
E610		AND 10	A:= 10 if '5' pressed, 00 otherwise.
47		LD B,A	
3EEF		LD A, EF	
DBFE		IN A, (FE)	Scan segment four of the keyboard.
E604		AND 04	A:= 04 if '8' pressed, 00 otherwise.
B0		OR B	
21B05C		LD HL, X_COORD	Prepare to adjust the X co-ordinate.
FE04		CP 04	
2809		JR Z, RIGHT	Jump if '8' pressed, but not '5'.
FE10		CP 10	
2008		JR NZ, UP/DOWN	Jump unless '5' was pressed without '8'.
CD????	LEFT	CALL INC_COORD	Increment the X co-ordinate. (Note that the window itself moves right, hence the maze moves left.)
1803		JR UP/DOWN	
CD????	RIGHT	CALL DEC_COORD	Decrement the X co-ordinate.

THROUGH THE SQUARE WINDOW

```
10 PLOT 183, 103 Construct a frame for
                 the window
20 DRAW 0, 65
30 DRAW 65, 0
40 DRAW 0, -65
50 DRAW -65, 0
60 LET L=USR ????? This refers to the
                   label START
70 GO TO 60
```

USER DEFINED GRAPHICS

Type this Hex into the User Defined Graphics area of your machine.

Graphic	A	00	00	00	00	00	00	00	00
Graphic	B	01	01	01	01	01	01	01	01
Graphic	C	80	80	80	80	80	80	80	80
Graphic	D	81	81	81	81	81	81	81	81
Graphic	E	FF	00	00	00	00	00	00	00
Graphic	F	FF	01	01	01	01	01	01	01
Graphic	G	FF	80	80	80	80	80	80	80
Graphic	H	FF	81	81	81	81	81	81	81
Graphic	I	00	00	00	00	00	00	00	FF
Graphic	J	01	01	01	01	01	01	01	FF
Graphic	K	80	80	80	80	80	80	80	FF
Graphic	L	81	81	81	81	81	81	81	FF
Graphic	M	FF	00	00	00	00	00	00	FF
Graphic	N	FF	01	01	01	01	01	01	FF
Graphic	O	FF	80	80	80	80	80	80	FF
Graphic	P	FF	81	81	81	81	81	81	FF

DATA FOR MAZE

It's worth getting a friend to help you type this lot in!

```
53 9C 94 9C 94 9C 9C 9C
9C 9C 9C 9C 9C 9C 94 94
9C 9C 9C 9D 96 9C 9C 9C
9C 9C 9C 9C 9C 9C 9C 9D
93 97 93 97 92 9C 9C 9C
9C 9C 9C 9C 9C 95 93 93
96 9C 9C 9C 98 9C 9C 9C
9C 9C 9C 9C 9C 9C 9C 95
92 98 99 93 9A 9C 9C 9C
9C 9C 9C 9C 95 93 93 93
93 96 9C 9C 9C 9C 9C 9C
9C 9C 9C 9C 9C 9C 95 93
93 9E 9C 98 9C 9C 94 9C
9C 9C 9C 9D 93 93 93 93
93 93 96 9C 9C 94 9C 94
9C 9C 9C 9C 9C 95 93 93
92 9C 9C 9C 9C 9D 93 9E
9C 9C 9C 9C 99 93 93 93
93 93 9A 9C 95 93 97 93
96 9C 9C 9C 95 93 93 93
93 9E 9C 9C 9C 9C 98 9C
9C 9C 9C 9C 9C 91 93 93
93 9A 9C 95 93 93 93 93
93 96 9C 95 93 93 93 93
92 9C 9C 9C 9C 9C 9C 9C
9C 9C 94 9C 95 93 93 93
92 9C 9C 99 93 93 93 93
93 93 97 93 93 93 93 93
93 9E 9C 9C 9C 9C 94 9C
9C 9D 93 97 93 93 93 93
93 9E 9C 9C 99 93 93 93
93 93 93 93 93 93 93 93
92 9C 9C 9C 9C 95 93 9E
9C 9C 99 93 93 93 93 93
9A 9C 9C 9C 9C 99 93 93
93 93 93 93 93 93 93 93
9A 9C 9C 9C 9D 93 93 96
9C 9C 9C 99 93 93 93 92
```

CODE		ASSEMBLER	COMMENTS
3EEF DBFE	UP/DOWN	LD A,EF IN A,(FE)	Scan segment four of the keyboard.
E618		AND 18	A:= 08 if '7' pressed, 10 if '6' pressed, 18 if both '7' and '6' pressed, or 00 if neither key is pressed.
21B15C		LD HL, Y_COORD	Prepare to adjust the Y co-ordinate
FE10 2809		CP10 JR Z,DOWN	Jump if '6' pressed, but not '7'.
FE08 2008		CP 08 JR NZ,WINDOW	Jump unless '7' pressed without '6'.
CD???? 1803	UP	CALL INC_COORD JR WINDOW	Increment the Y co-ordinate.
CD????	DOWN	CALL DEC_COORD	Decrement the Y co-ordinate.
2A845C E5	WINDOW	LD HL,(DF_CC) PUSH HL	Stack PRINT position.
2A885C E5		LD HL, (S_POSN) PUSH HL	Stack PRINT co-ordinates.
AF		XOR A	A:= 00.
323C5C		LD (TVFLAG), A	Print to the upper part of the screen.
FD365538		LD (ATTR_T), 38	Set colours to black on white.
2AB15C		LD HL, (Y_COORD)	L:= Y co-ordinate of the window.
67		LD H,A	HL:= Y co-ordinate of the window.
112000		LD DE,Length	DE:= size of the maze.
CDA930		CALL 30A9, HL=HL*DE	HL:=(Y co-ordinate) *(size of maze).
ED5BB15C		LD DE, (X_COORD)	E:= X co-ordinate of the window.
57		LD D,A	DE:= X co-ordinate of the window.
19		ADD HL,DE	HL:= square number of the window position.
11????		LD DE,MAZE	Point DE to the start of the maze.
19		ADD HL,DE	Point HL to the appropriate square.
110108		LD DE,0801	D:= count (number of rows to print), E:= PRINT AT co-ordinate required.
3E16 D7	LOOP_1	LD A,"at_ctrl" RST 10	PRINT AT
7B D7		LD A,E RST 10	required row,
3E17 D7		LD A,17 RST 10	17h;
0608		LD B,08	B:= number of columns in the window.
7E	LOOP_2	LD A,(HL)	A:= next character from the maze.
23		INC HL	Point to the next character.
D7		RST10	Print this character in the window.
10FB		DJNZ LOOP_2	Print the whole row.
0E18		LD C,Length-8	BC:= displacement to the next row of the window.
09		ADD HL,BC	Point HL to the next row.
1C		INC E	Increment PRINT AT position.
15		DEC D	
20EA		JR NZ,LOOP_1	Repeat for all rows.
E1 22885C		POP HL LD (S_POSN), HL	Restore PRINT co-ordinates.
E1 22845C C9		POP HL LD (DF_CC),HL RET	Restore PRINT position.

THROUGH THE SQUARE WINDOW

```
9C 9C 9C 9C 9C 9C 99 93
93 93 93 93 93 93 9B 93
9E 9C 9C 9C 9C 91 9A 91
96 9C 9C 9C 99 93 93 9B
96 9C 9C 9C 9C 9C 9C 99
93 93 93 93 93 92 9C 99
96 9C 9C 9C 9D 9A 95 93
93 96 9C 9C 9C 99 9A 9C
99 96 9C 9C 9C 9C 9C 9C
99 93 93 93 93 93 96 95
93 96 9C 9C 95 97 93 93
93 9A 9C 9C 9C 9C 9C 9C
9C 91 96 9C 9C 9C 9C 9C
9C 99 93 93 93 93 9B 93
93 93 9E 9C 99 93 93 93
92 9C 9C 9D 96 9C 9C 9C
9D 93 93 96 9C 9C 9C 9C
9C 9C 91 93 93 92 9C 91
93 93 96 9C 9C 99 93 93
9A 9C 9C 9C 91 9E 9C 9C
95 93 93 93 96 9C 9C 9C
9C 95 93 93 93 93 97 93
93 93 93 9E 9C 9C 99 9A
9C 9C 9C 95 9A 9C 9C 95
9A 99 93 93 93 9E 9C 9C
95 93 93 93 93 93 93 93
93 93 92 9C 9C 9C 9C 9C
9C 9C 9C 98 9C 9C 95 9A
9C 94 99 93 9A 9C 9C 9D
93 93 93 93 93 93 93 93
93 93 93 96 9C 9C 9C 9C
9C 9C 9C 9C 9C 95 92 9C
9D 93 9E 98 95 96 9C 9C
99 93 93 93 93 93 93 93
93 93 93 93 96 9C 9C 9C
9C 9C 9C 9C 95 93 93 9E
9C 98 9C 95 93 93 96 9C
9C 99 93 93 93 93 93 93
93 93 93 93 93 96 9C 9C
9C 9C 9C 95 93 93 9A 9C
9C 9C 95 93 93 93 93 96
9C 9C 99 93 93 93 93 93
93 93 93 93 93 93 96 9C
9C 9C 95 93 93 92 9C 9C
9C 95 93 93 9A 99 93 93
96 9C 9C 99 93 93 93 93
93 93 93 93 93 93 93 96
9C 95 93 93 93 93 96 9C
95 93 93 9A 9C 9C 99 93
93 96 9C 9C 99 93 93 93
93 93 93 93 93 93 93 93
97 93 93 93 93 93 93 97
93 93 9A 9C 9C 9C 9C 99
93 93 96 9C 9C 99 93 93
93 93 93 93 93 9A 98 99
93 93 93 93 93 9B 93 93
93 9A 9C 9C 94 9C 9C 9C
99 93 9B 96 9C 9C 99 93
93 93 93 93 9A 9C 9C 9C
99 93 93 93 92 9C 91 93
9B 96 9C 9C 99 9E 9C 9C
9C 98 9C 91 96 9C 94 99
93 93 93 9A 9C 9C 9C 9C
9C 99 93 93 93 97 93 92
9C 99 9E 9C 9C 9C 9C 9C
9C 9C 95 93 93 97 93 96
93 93 9A 9C 9C 9C 9C 9C
9C 9C 99 93 9B 93 93 93
96 95 96 9C 95 9E 9C 9C
9C 95 93 9B 93 93 92 91
93 92 94 94 9C 9C 9C 9C
9C 9C 9C 98 9C 99 93 93
93 9B 93 97 92 9C 9C 9C
95 93 9A 9C 99 93 93 9B
9C 9C 9A 91 9C 9C 9C 9C
9C 9C 9C 9C 9C 9C 99 93
93 96 99 92 90 9C 9C 95
93 9A 9C 9C 9C 98 98 9D
93 93 97 93 93 9E 9C 9C
9C 9C 9C 9C 9C 9C 9C 99
93 93 96 99 92 9C 95 93
9A 9C 9C 9C 9C 9C 9C 95
93 9A 99 93 9A 9C 9C 94
9C 9C 9C 9C 9C 9C 9C 9C
99 93 93 9E 99 97 93 9A
9C 9C 9C 9C 9C 9C 95 93
9A 9C 9C 98 9C 9C 9C 98
9C 9C 9C 9C 9C 9C 9C 9C
9C 99 9A 9C 9C 99 9A 9C
9C 9C 9C 9C 9C 9D 9B 46
```

YS

ADVERTISEMENT

fantasy SOFTWARE

Fantasy Software, Fauconberg Lodge, 27A, St.George's Road, Cheltenham, Glos GL50 3DT

NEWSLETTER No.1

PAUL'S BIT

Well here it is! The long awaited newsletter. Most of you should have already received a copy of your copy of our newsletter via your trusty postman. However we have decided to print a newsletter in the pages of YOUR SPECTRUM on a regular basis. Members will still receive the occasional letter or so in the post from time to time. Well in case you're wondering what we've been up to this past couple of months, we have been working flat out on our new game "BEAKY AND THE EGGSNATCHERS" to be released shortly on the 48K Spectrum and also the Commodore 64. I can remember in the last newsletter telling you that you were going to like "THE PYRAMID" and it appears from the success of the game that I was right. I now find myself in a similar position with "BEAKY AND THE EGGSNATCHERS". Having seen the game develop screen by screen in the skillful hands of Bob Hamilton, the pure attention to detail on graphics, the humourous game plan and excellent playability adds up to a thoroughly recommended game. I will send you all the details by post as soon as the the game is finished.

We have some exciting things planned for September with games design already underway. We are hoping to get the official licence to produce a game on a very will known theme – enough said eh!

Anyone wishing to join the Micro Club complete the form below and return to us at the address shown.

A short note on high score entries – anyone who fails to quote their membership number and code will not be entered into the office computer and hence you run the risk of being deleted! Please print your codes clearly because if they can't be read they can't be verified, if your name is missing from our high score lists this is probably why.

QUICK QUIZ by Paul

What was the highest position obtained by "THE PYRAMID" in the W.H.SMITH chart recently? First 10 correct entries receive a free copy of "BEAKY AND THE EGGSNATCHERS" on release. Entries on a postcard please and quoting membership number!

BOB'S BIT

Firstly I would like to take this opportunity to thank everyone for the many, many super letters I've received since the release of "The Pyramid" back in October. I must apologise for not being able to answer all your letters personally but I'm afraid the sheer volume makes it very difficult (I'd never get any software written!). They are however very much appreciated.

To put an end to much rumour and speculation my latest game will not feature Ziggy (sorry fans, but he's having a very well earned rest until later in the year) but our new star, Beaky who I'm sure you will love even more than Ziggy.

'Beaky and the Eggsnatchers' has taken a long time to develop and I hope you will find it as enjoyable and challenging to play as I found it to write. I'm too modest to say much about it (that is Paul's department) except that I think you'll really enjoy the graphics and find it just as addictive as "The Pyramid".

It seems that completing Doomsday Castle is proving a formidable task with only 5 people on record as having successfully saved the Universe from the evil Scarthax and escaped to join the roll of honour. Congratulations to Russell Capel who was the first to complete the task and that was within a week! The Pyramid number puzzle is also proving formidable. The two £50 cash prizes are still waiting to be won so go and get puzzling.

Finally, you can tell your Commodore owning friends that "The Pyramid" is now available for them as well, with no less than 120 different aliens and many with intelligence – get them to buy it so you can have a go!

PYRAMID TOP 50

1	P Harkins, Cleveland	137739
2	Giles Ahern, Surrey	137499
3	Alastair Douglas, Belfast	137071
4	Joanne Thompson, Birkenhead	136731
5	Scott Hamilton, Lanarkshire	136616
6	Graham Phillips, Wiltshire	136233
7	Colin Maclean, Ross-shire	136116
8	Stephen Lea, South Yorks	136022
9	Simon Hawkins, Cheltenham	135999
10	Michael Crowe, Birmingham	135934
11	Susanne Dodwell, South Glamorgan	135929
12	Russell Capel, Cheltenham	135877
13	Andrew Moss, Middlesex	135850
14	Aron Gadd, Hampshire	135827
15	David Baxter, Lanarkshire	135715
16	Hamish Overend, Buckinghamshire	135116
17	Martin Angus, Scotland	135111
18	Mark Young, Berkshire	135105
19	Doug Jefferson, York	134848
20	John Hicks, West Glamorgan	134716
21	Mrs R M Foss, Manchester	134538
22	Mark Rumble, South Glamorgam	134409
23	Les Gibbins, Devon	134210
24	David Priddle, Cardiff	134191
24	Mark Graham, Edinburgh	134067
25	Mark Graham, Edinburgh	134067
26	George Price, Scotland	133927
27	R O Hankinson, Cheshire	133926
27	William Hill, Berwickshire	133926
29	Robert Cruden, Aberdeenshire	133769
30	Owen Whitehead, Barnsley	133653
31	Mr D Alden, Essex	133498
32	Andrew Mackinnon, Warwickshire	133409
33	Anthony Lee, Kent	132867
34	Edward Miliband, London	132856
35	David Sneddon, Scotland	132758
36	Paul Fiarweather, Merseyside	132277
37	Ashley Newton, Essex	131998
37	Neil Petherick, Hertfordshire	131998
39	A Vazquez, Lancashire	131950
40	Andrew Kite, Yorkshire	131971
41	R Goodman, Margate	131607
42	Alan Best, Scotland	131437
43	Jason Weir, Bristol	131358
44	Ian Murrfitt, Kent	131222
45	P Westaway, Surrey	130914
46	Andrew Sanders, Isle of Man	130853
47	Robert Burgess, South Yorkshire	130526
48	Andrew Jones, Worcester	130501
49	Steven Busby, Essex	130481
50	Elizabeth Bennett, Cambridgeshire	130364

MEMBERSHIP FORM

Please return to: Fantasy Software, Fauconberg Lodge, 27A, St.George's Road, Cheltenham, Gloucestershire GL50 3DT

Name

Address

Postcode

I own a Spectrum ☐ I own a CBM 64 ☐ Please tick as appropriate

JOYSTICK DIY

In our first issue, Stephen Adams presented his recipe for a DIY Speccy keyboard buffer. This month, under the guise of further expansion, he details how you can add Sinclair-type joysticks to the board.

The way that Sinclair Research has designed its joystick sockets on the Interface 2 unit is for the sticks to simulate the number keys; so '1' to '5' are operated by Joystick One and '6' to '0' by Joystick Two. Let's look at the first diagram and see just what's inside the standard Atari joystick.

You'll find a set of five switches joined together by a 'common' wire. Four of them are controlled by the direction of the joystick and the fifth is the Fire button. Usually, only one of the direction buttons is in operation at any one time.

ASSEMBLE MODE

Once you have your buffer board, all you'll need to buy are two nine-way 'D' type sockets — the ones which are standard for all Atari joysticks. First, wire up the address line (A11 or A12) to the common on the Atari socket (pin eight). You can get this from the spare hole beneath the IC socket. As a result, a binary zero on the data lines connected to the other sides of the switches can only occur when the number keys are being checked.

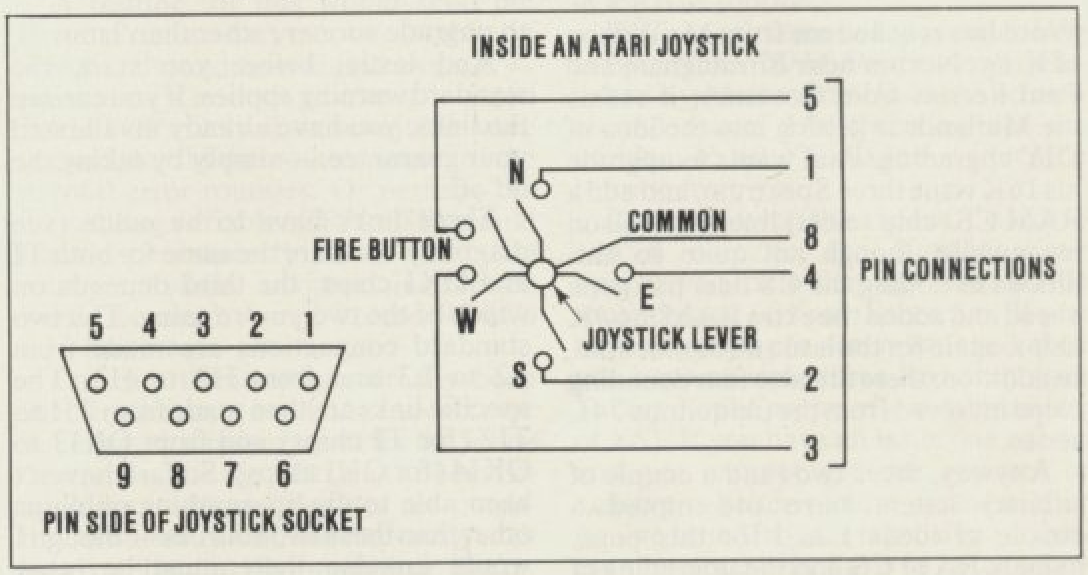
The data lines must then be connected to the 'North', 'South', 'West', 'East' and 'Fire' connections on the socket. The socket is shown here from the pin side; there's plenty of room above the two ICs to mount the sockets, having first wired them into the board. The tracks to use are the ones above the straps which carry the data lines from one side of the board to the other. I would suggest, for single joystick operation, that you make use of the left-hand side joystick and that you secure the socket to the board with screws to make it easier to disconnect.

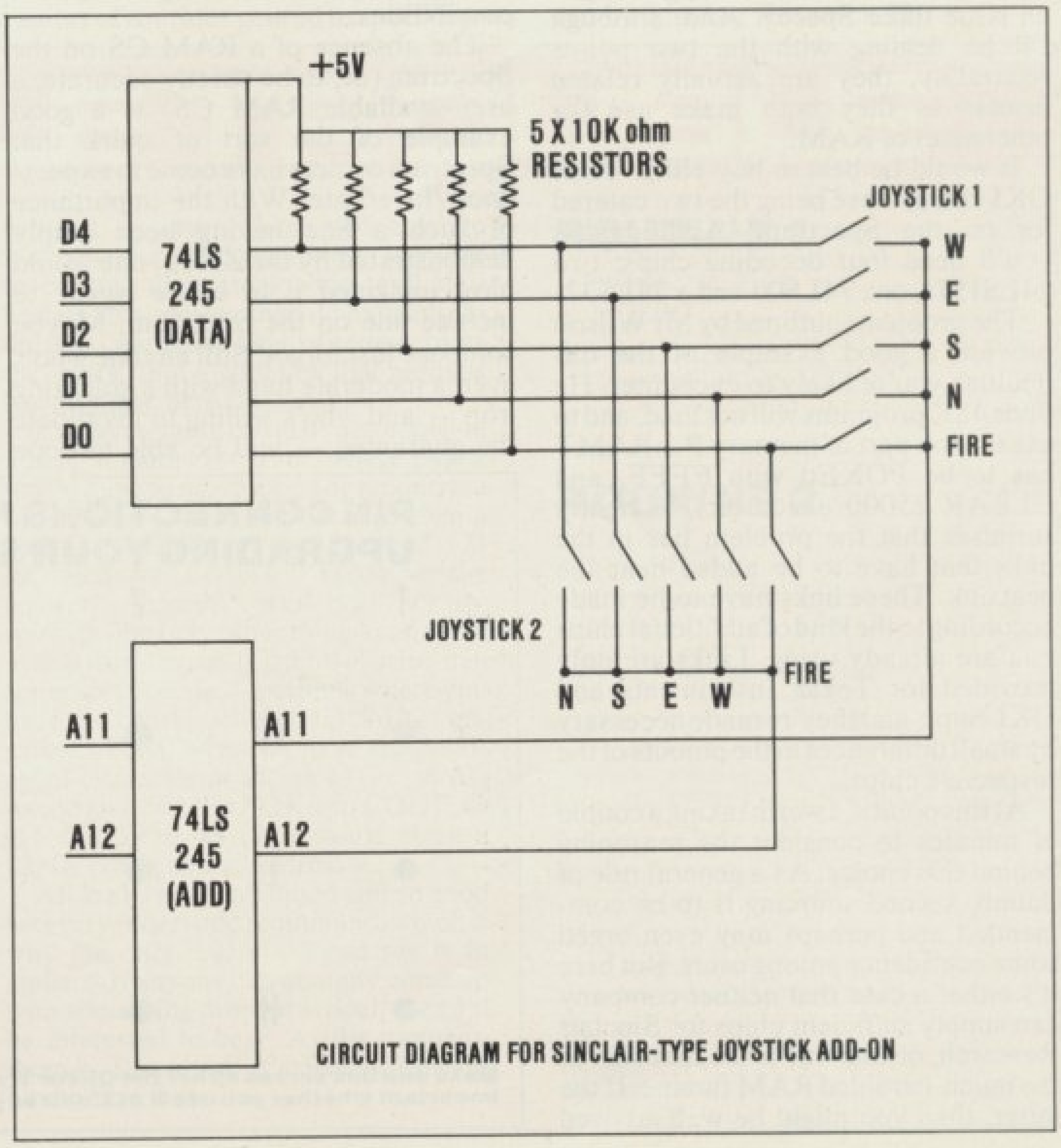
Each of the data lines must be connected to a 10K ohm resistor, so that the normal condition of the joysticks is binary one (+5 volts). This is done to counter the capacitance effect that can easily result from using long leads. It's also worth doing this if you find that the keyboard 'locks up' — because the correct binary one signal is not strong enough.

Any standard Atari joystick can then be plugged into the socket and the interface tested to see if it produces the correct numbers. If the numbers look wrong, check the connections to the joystick socket. An added extra is that the joystick may also be modified to fit any games which use a particular and specific set of five keys; just connect up the appropriate address line to pin eight.

The final bonus is that Psion and Sinclair Research have been writing games that use this interface for some time now; check the tapes you have already — you might be surprised to find that they have joystick commands already built-in. YS

WHAT YOU NEED:
5 x 10Kohm resistors
2 x 9-way 'D' type sockets (standard for all Atari joysticks)





CIRCUIT DIAGRAM FOR SINCLAIR-TYPE JOYSTICK ADD-ON

WHEN THE CHIPS ARE DOWN

YS's own reader-interactive workshop. Theory, philosophy, bugs and fixes — all is controlled and compiled by Speccy specialist Ian Beardsmore. Write to him c/o Spectrolysis, Your Spectrum, 14 Rathbone Place, London W1P 1DE.

Word has reached me from Mr Wilson of Kings Norton near Birmingham and Paul Ferries from Coventry; it seems the Midlands is getting into the idea of DIY upgrading. Paul wants to upgrade his 16K issue three Spectrum, and add a RAM CS (chip select) line; Mr Wilson meanwhile, though not quite so ambitious as adding the CS line, has gone ahead and added the extra RAM in 64K chips, again for the issue three machine. In addition, there are also four decoding chips involved from the ubiquitous 74L series.

Anyway, these two (and a couple of similar) letters have pre-empted a couple of ideas I had for this page, namely RAM CS and the upgrading of an issue three Speccy. And, although I'll be dealing with the two points separately, they are actually related insofar as they both make use (or otherwise) of RAM.

It would be best to buy either TI or OKI chips, these being the two catered for on the Spectrum. Additionally, you'll need four decoding chips: two 74LS157s, one 74LS00 and a 74LS32.

The problems outlined by Mr Wilson provide a good example of the difficulties you're likely to encounter. He finds 48K programs will not load, and to use the top part of memory P_RAMT has to be POKEd with FFFF, and CLEAR 25000 executed. He rightly surmises that the problem lies in the links that have to be added near the heatsink. These links have to be made according to the kind of additional chips you are already using. Links are only provided for Texas Instruments and OKI chips, and they're made necessary by small differences in the pinouts of the respective chips.

At this point it's worth taking a couple of minutes to consider the reasoning behind this choice. As a general rule of thumb, second sourcing is to be commended and perhaps may even breed some confidence among users. But here it's either a case that neither company can supply sufficient chips for Sinclair Research, or that this is the first sign of the much-heralded RAM famine. If the latter, then you might be well advised to upgrade sooner rather than later.

And lastly, before you start, the standard warning applies. If you can see the links, you have already invalidated your guarantee — simply by taking the lid off.

Three links have to be made (see diagram). Two are the same for both TI and OKI chips, the third depends on which of the two you're using. The two standard connections are made from L2 to L3 and from H2 to H3. The specific links are then made from TI1 to TI2 (for TI chips) and from OKI3 to OKI4 (for OKI chips). So far I haven't been able to dig up anything on chips other than these two sources — though I would imagine there must be other possibilities.

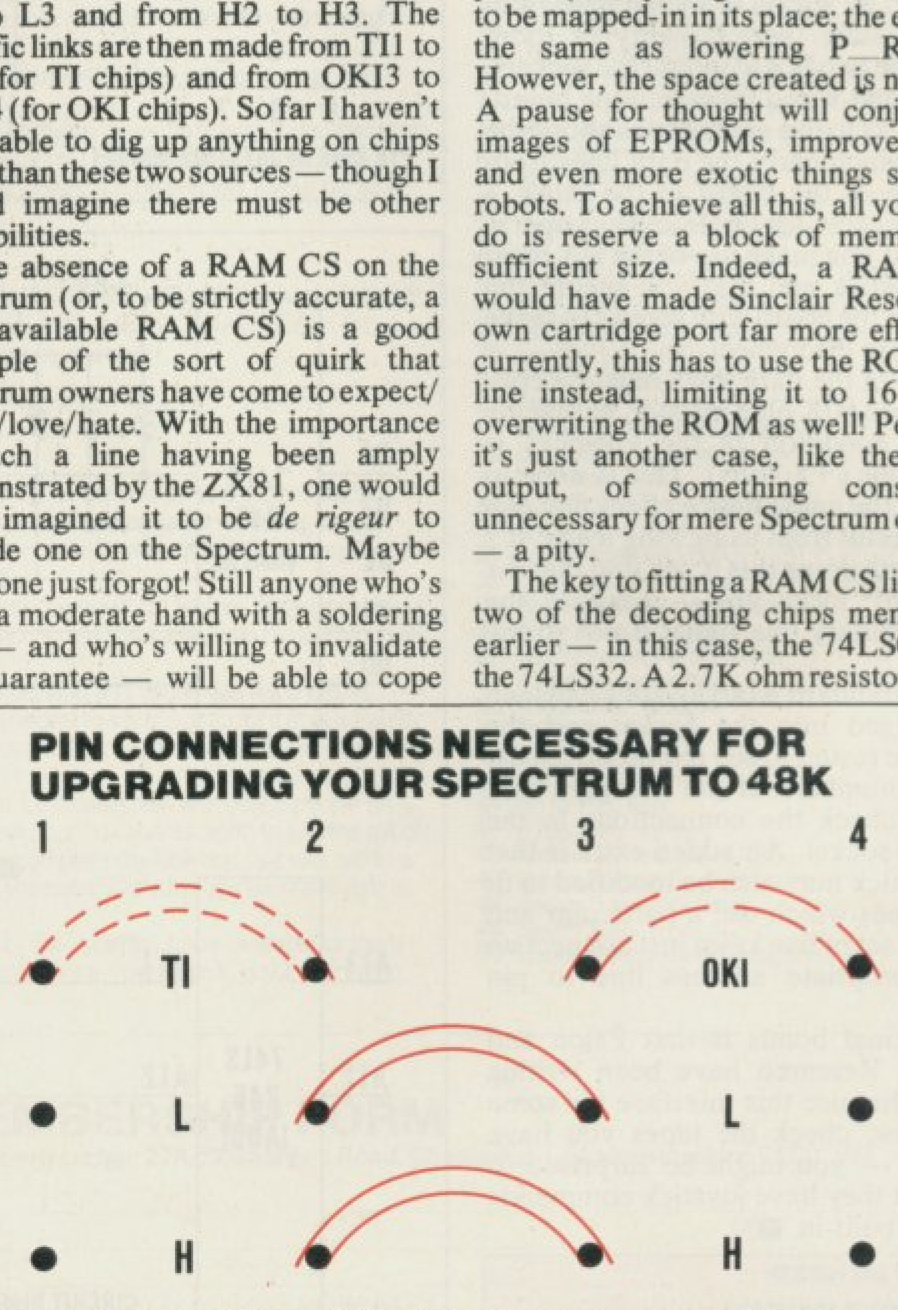
The absence of a RAM CS on the Spectrum (or, to be strictly accurate, a user available RAM CS) is a good example of the sort of quirk that Spectrum owners have come to expect/know/love/hate. With the importance of such a line having been amply demonstrated by the ZX81, one would have imagined it to be *de rigeur* to include one on the Spectrum. Maybe someone just forgot! Still anyone who's even a moderate hand with a soldering iron — and who's willing to invalidate the guarantee — will be able to cope with this quite modest modification.

In fact, all you need to do to get a RAM CS is to add a single resistor. The method for this was worked out early on in the life of the Spectrum by Stephen Adams, as part of his *Eve Adaptor* which can be used for interfacing ZX81 devices to the Speccy.

The RAM CS mod allows a block of memory to be mapped-out and a peripheral (or anything else for that matter) to be mapped-in in its place; the effect is the same as lowering P_RAMT. However, the space created is not lost. A pause for thought will conjure up images of EPROMs, improved I/O, and even more exotic things such as robots. To achieve all this, all you need do is reserve a block of memory of sufficient size. Indeed, a RAM CS would have made Sinclair Research's own cartridge port far more effective; currently, this has to use the ROM CS line instead, limiting it to 16K and overwriting the ROM as well! Perhaps, it's just another case, like the video output, of something considered unnecessary for mere Spectrum owners — a pity.

The key to fitting a RAM CS lies with two of the decoding chips mentioned earlier — in this case, the 74LS00 and the 74LS32. A 2.7K ohm resistor needs

PIN CONNECTIONS NECESSARY FOR UPGRADING YOUR SPECTRUM TO 48K



Make one link across either the OKI or TI and a second across either H or L. It's not important whether you use H or L unless you're using Sinclair chips.

WHEN THE CHIPS ARE DOWN

to be soldered across from pin eight of the 74LS00 chip, to pin 10 on the 74LS32 chip; naturally, the track on the PCB has to be cut as well. Now, a line from pin 10 to position four on the expansion port (next to the slot) becomes a user available RAM CS. When in use it needs to be kept high, though it will only come into play when an address is called within the reserved block. The reserved block will always be at the higher end of RAM, but can be up to 32K long. Clearly, a 48K Spectrum will have be used, and trying to put the line on the lower 16K is tricky as you run the risk of overwriting all the essential reserved area at the lower end of the RAM.

STACK SITUATIONS

Thank you for all your letters that are now starting to arrive; but although there's lots of interesting ideas, I seem to be suffering some sort of hardware hijack — this isn't meant to be just a hardware column! And for those of you who have sent in Microdrive material, sorry but my Interface 1 unit is currently doing its own impression of Sinclair Research's QL delivery dates when it comes to readability — never mind, I'm sure it will be 'ready soon'.

But there is one idea I'd like to throw at you — mainly because it was thrown at me recently. There's been stuff published about using the Microdrive vector as a way in to adding your own commands to Spectrum's own Basic. And there's also been discussion about using the Stack Error Pointer to point to a routine that will NEW a program if someone tries to break in. So . . . why not use the same Stack Error Pointer to send you a routine, or suite of routines, of your own. I've only mulled the idea over so far — without making much progress, I must admit.

A routine for this would send the Error Pointer to your own table of commands and associated syntax — presumably in code above a lowered RAMTOP. Only if these criteria were not met would the pointer return to the normal error routines. Or perhaps the controlling variable (ERR_SP) would remain permanently set at your own routines, while another instruction would be used to call up the ROM's own routines.

I'm sure the idea has already been used by the pros, but I can't recall ever having seen anything on it for the home user — despite the fact that the benefits could be immense. Besides the potential of suites of utilities featuring such handy things as *Renumber* or *Trace*, suites of routines geared to specific uses would also be possible; *Fill* on a graphics collection would be an obvious example here.

Anyway, to finish this month here is something from Stephen Smart of Pollock, Glasgow (a man who has the intellect to describe himself as a 'pleased reader'):

"If you go into extended mode and press a number that represents a colour — for example, '5' for cyan — and follow this with a space, what you'll get on the screen is a square of that colour. In Basic programs, colours can be specified like this, REM statements highlighted and text made generally more clear. But how can I find out where the routines for drawing circles and lines in ROM are? I'd like to use them amended in a RAM routine."

Well, as a first move I suggest looking at *Spectrum ROM Disassembly* published by Melbourne House. And as a second move, I'm throwing out the problem to all *YS* readers — let's be hearing from you.

Lastly, to wrap this month, I'd like to add a note to the *Tuning Your Spectrum* article that appeared in the second issue of *Your Spectrum*. If you haven't got the pre-sets mentioned, then you have an issue three Spectrum (see *Sexing Your Spectrum* in the third issue of *YS*). If you have an issue one or two and you cannot get the pre-sets to achieve a better picture — even after you've tuned in your television — then the computer will have to be checked out by Sinclair Research. YS

DOUBLE TROUBLE

Here are two routines from Mr K Yeoman of Poole in Dorset. The first shows one of the interesting vagaries of the Speccy's STRING$ command and the second is an attempt to make use of it. Unfortunately, he says the attempt is less than inspired, and it probably offers no advantage at all over using a more conventional string. He says, "I was trying to look at the idea of some sort of sprite-type system that might be possible if the sprite was symmetrical". (SCREEN$ can be made to read user-defined graphics, by judicious POKEing of the system variables.)

Personally, I think it's only fair that if Sinclair Research is going to program 'quirks' into the Speccy, then users should go out of their way to find other non-anticipated uses for them. Anyway, my first thought was to find a way of simulating the DUPLICATE command of Forth, where one 'copy' of the number could be used, while the other copy remained. However, Basic is not Forth, and it doesn't have anything like the direct access to the stack — it probably wouldn't be necessary. But I'm sure that the writers of some of the exciting (?) games that infest certain computing magazines will find a way of using it — ones where you have to manoeuvre around a shape that happens to be two characters wide, and if it bashes into another then it changes into the new character. This, of course, turns the 'quirk' into a gimmick that perhaps has no place here (???).

However, there could be a use for it in a simple maths program, where pupils would have to move a cursor around the screen to a certain number, and then a line like LET b = a$ (1) + a$ (2) followed by PRINT "What is the answer". Hmm . . . still not very inspiring! The only other thing I can think of that might come in useful is using it in some sort of code — perhaps where you have the character 'e' (CHR$ 101) on-screen. Then, when we have the duplicated character in a$, a$ (1) + a$ (2) would give 202. So POKE n(CODE a$ (1) + CODE a$ (2)) would store a LINE command at address n.

All in all *very* convoluted and no good for every other (odd) command, which is why the only real use I can see is in ciphers. If anyone can actually come up with something more practical, then I'd be interested to hear. At the moment, though, I'm afraid it's Round One to Uncle Sir Clive.

ROUTINE 1

```
1 PRINT "*";
2 LET a$=""
3 LET a$=a$+SCREEN$ (0.0)
4 PRINT " ";a$
5 STOP
```

ROUTINE 2

```
10 LET x=10
20 LET y=10
25 GO SUB 1000
35 PRINT AT x.y;"#"
40 LET a$=""
45 LET a$=a$+SCREEN$ (t.z)
47
60 IF INKEY$="0" THEN  STOP
70 IF INKEY$="1" THEN  GO TO 100
80 IF INKEY$="2" THEN  GO TO 200
90 IF INKEY$="" THEN  GO TO 60
99
100 LET w=w-1
120 PRINT AT v.w;" ";a$;" "
140 GO TO 60
190
200 LET w=w+1
220 PRINT AT v.w;" ";a$;" "
240 GO TO 60
990
020 LET v=x: LET w=y
040 LET t=x: LET z=y
060 RETURN
```

SUBCONSCIOUSNESS



Fantasise over features . . .



Marvel at machine code . . .



Revel in reviews . . .



Lust over listings . . .

If you recognise any of the critical conditions portrayed above, then you're probably a lost cause as far as computing goes — admit it, you're hooked! And, as a devout follower of Sir Clive's wonder machines, you don't want to miss out on one of the best cures around (our magazine carries no Government health warning).

You've doubtless heard the rumour that you'll always be able to take a stroll down to your local newsagents and pick up a copy of *YS*, no problem? But the fact is that *Your Spectrum* issue one sold out so fast, we're still getting calls from people anxiously pulling their hair out trying to secure a copy. Really!

So, that's why this month, we're kicking off our *Speccy Subs* service. For just £12 (in UK and Northern Ireland only — it's a little bit more for Europe and Air Mail) you can guarantee that each issue will come thudding through your door each and every month.

All you have to do is fill in the attached coupon (or a photocopy of same) and unlock your chequebook. Of course, if you're a credit card king, just give us your number and we'll do the rest — that'll do nicely.

YS PRIORITY ORDER FORM

I would like to subscribe to *Your Spectrum and QL User* for the next 12 issues, starting with the .... issue.

I enclose a cheque/postal order made payable to Sportscene Specialist Press Ltd for (tick the appropriate box):

☐ £12 (UK and Eire)
☐ £15 (Europe) ☐ £25 (Airmail)

Or please charge my Access/Visa/Diners/American Express card (please delete where applicable).

Account Number ____________

Name: ____________

Address: ____________

____________ Postcode

(please use block capitals)

Send the completed coupon plus payment to **Speccy Subs, Your Spectrum, 14 Rathbone Place, London W1P 1DE.**

If you don't want to deface your magazine please make a photocopy of this order form.

MUSIC ON SPEC

Sweet sounds and Spectrums aren't usually words uttered in the same breath! However, the program shown here, courtesy of Sam & Simon Goodwin, allows you to connect any Spectrum to the light-pen port of a Casio synthesiser, giving you a versatile new peripheral.



IAN McKINNELL

The best news of all is that the entire program to carry out this wondrous interfacing task is only about a dozen lines long (see the listing further on). The program was tested using a Casio VL-5 polyphonic synthesiser and a 16K Spectrum with Interface 1 connected. With this configuration no extra hardware is needed, other than a couple of plugs and some wire. It should be possible to use a cassette recorder or amplifier in place of Interface 1 if you've still to purchase (or receive!) the interface.

REFLECTIONS

The Casio light-pen produces a pulse of electricity whenever it passes over a reflective surface. So, armed with the catalogue of tunes supplied with the synthesiser, the first problem was to work out the pattern used to transfer information.

It turned out that tunes are stored in eight-bit bytes, each composed of two four-bit sections. Pitch information is stored separately from the rhythm data — you load the frequencies of notes to be used independently from the durations. Because of the four-bit basis for the information, we chose to use hexadecimal (base 16) to represent values sent to the synthesiser.

The synthesiser can tell pitches from rhythms by looking at the start of the data — the hexadecimal sequence '0408' prefixes the pitch information, and '0404' marks the rhythm data. There is scope for further experimentation here since we suspect that larger synthesisers use the other values to represent chords — the VL-5 only allows you to load monophonic music (although you can play along with it once it is loaded).

Bar coded music is normally stored as a number of musical 'bars', each with its own start and end marker. This is convenient since it means that Casio doesn't have to print enormously wide books with bar-codes on them — they can split a tune up over more than one line. However, the Spectrum doesn't need to print the music and, consequently, the entire tune entered can be transferred as a single bar. You could use multiple bars by altering the third digit of the start marker. For example, 0414 marks the second bar of rhythm information, and 04A8 would prefix the eleventh bar of melody. The Casio uses an 'extra' bit at the end of each bar to indicate whether or not there is another bar to come, so that must also be set appropriately.

MAKING NOTES

Each pitch is represented by two hexadecimal digits. The first digit is the note number in the octave: a value from one to 12 (Hex 'C'). The *Note Codes* table summarises the relationship between this digit and the note produced. The value zero represents a 'rest' (silent pause). The Hex digit 'D' (13) tells the synthesiser to switch on the automatic rhythm generator.

The second digit associated with each pitch tells the computer the octave number of the note. Use the value zero if no octave number is needed — eg. for a rest. Most of the pre-supplied sounds allow four different octaves to be specified, using the values '4' through '7' but 'flute', 'violin' and 'pretty', can only be generated over a three octave range — the value '7' always produces a top C note if specified for one of those sounds. The use of values between '4' and '7' hints at other capabilities of the format — presumably, further octave values can be selected on synthesisers more versatile than the VL-5.

As a simple example, to specify two notes an octave apart in pitch, you can type '04081617'. Once you press Enter, the information is automatically converted and transmitted to the syn-



With the vast range of games now available for home computers each holds a different challenge. Now there are three Competition-Pro Joysticks in the Kempston range that will come to your aid and improve your playing capabilities. Your movements will become quicker, with finger poised over the fire button at the ready, your scores will be out of this world. Playing will be so much easier! If you're serious about the challenge your computer presents then you shouldn't be without a Kempston Competition-Pro Joystick.

NEW

PRO 1000

Ergonomically designed handle and base ⋆ 8-way arcade quality leaf switches for precision control ⋆ 5 foot cable to allow more comfortable playing positions ⋆ Rubber return for smooth control ⋆ Rubber protector pads on base ⋆ Base fire bar for left or right hand control ⋆

NEW

PRO 3000

All the winning features of the Pro 1000 plus top fire button for precision one handed play ⋆ trigger fire button for true arcade realism ⋆

PRO 5000

Again all the winning features of the Pro 1000 but incorporating true craftsmanship plus large 1¼" control knob for comfortable play ⋆ Unique tapered shaft for maximum sensitivity ⋆ Dual independent fire buttons ⋆ Nylon covered steel shaft for maximum robustness

KEMPSTON JOYSTICK INTERFACE FOR THE ZX SPECTRUM

Allows the use of any Atari/Commodore type joystick with the popular ZX Spectrum computer. With well over 100 games on the Spectrum now compatible with the joystick this is a must for any Spectrum owner who wants the power of a joystick added to the power of the Spectrum. Simply plugs into the edge connector at the back of the Spectrum. No modifications necessary.

Some games you can buy were produced before Joysticks were available for the ZX Spectrum. With the aid of Kempston conversion tapes these games can now be played with any of the Kempston range of joysticks.



CONVERSION TAPE I
– Converts: Horace Goes Skiing, Hungry Horace, Flight Simulation (Sinclair), Arcadia, Penetrator, Spectres.

CONVERSION TAPE II
– Converts: Orbiter, Escape, Tanx, Centipede, Spectral Invaders, Cruising on Broadway, Frenzy.

CONVERSION TAPE III
– Converts: Ah Diddums, Jumping Jack, Molar Maul, Androids, Horace And the Spiders, Space Raiders, Winged Avenger, Muncher.

ALL £4.95 each

TRADE
ENQUIRIES
WELCOME

Most items are available from W H Smith, Boots, Spectrum Computer Centres and good computing shops or direct from:

KEMPSTON MICRO ELECTRONICS LTD

Unit 30 Singer Way, Woburn Road Industrial Estate, Kempston, Bedford, MK42 7AF. Tel: (0234) 856633 Telex: 826078 KEMPMI G

MUSIC ON SPEC

thesiser — you'll see the display border flash as the transfer takes place. The VL-5 then plays a short musical 'jingle' to signal that the information was received without error; a low note sounds if there's something wrong with it.

The end of a bar of either pitch or rhythm data is marked by a Hex 'F' digit, followed by a single digit 'checksum'. Both are automatically produced by the program, so you don't need to type the end marker when you enter the pitch or rhythm data.

EASY RHYTHM

The rhythm information is likewise transferred as a sequence of eight-bit values. After the prefix '0404', subsequent pairs of Hex digits represent 240 possible note durations. Only 240 values are allowed since any value starting with the Hex character 'F' is treated as an end-marker. This wouldn't matter much, were it not for the fact that the values are stored 'backwards' — the first digit represents the units and the second the multiples of 16! This means, for example, that you can represent the durations '0' to 'E' (14 decimal), but not 15 which would be sent as 'F0'. Likewise, the values 31, 47 and so on, are also forbidden.

The range of durations is ample for most music — a value of 134 (Hex '68') produced a note lasting 13 seconds on the VL-5 when it was set to slow tempo. Of course, the slider control can be used to make fine alterations in the tempo of music once it has been transferred. The value '00' produced a note so short that it was barely perceptible.

THE PROGRAM

The program listed by *Sam Goodwin* was written for speed and conciseness rather than legibility. Thus, annotation is provided for those who wish to adapt it — perhaps to produce computer-generated music or even a full-scale synthesiser control program.

WIRING UP

The diagram shows the wiring needed to connect your Interface 1 to a Casio VL-5. The socket on the Casio is a rather odd variety 4-pin DIN, but that shouldn't matter too much since most of you will, no doubt, simply poke the wires into the socket and trust to luck! For your information, though, the plug is available for a few pence from Maplin Electronic Supplies, catalogue number HH26D.

Make sure that the signal conductor from the Spectrum — the very end of the jack plug — goes to the small socket of the DIN plug, rather than the shield. The diagram of the DIN plug assumes that you're soldering it with the pins pointing away from you.

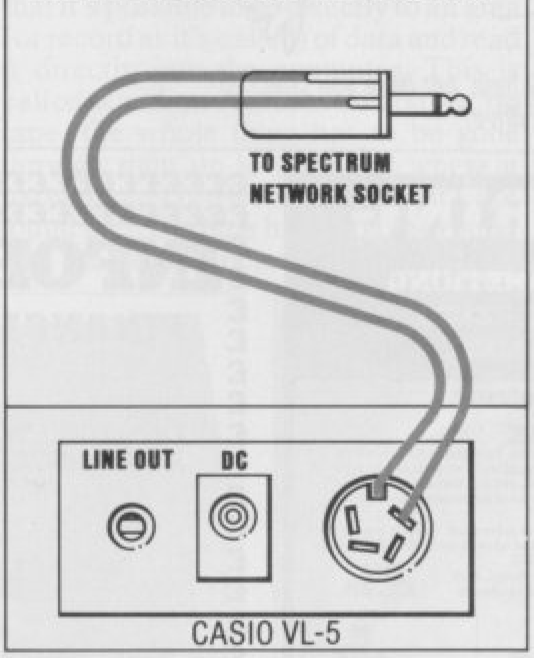
The connector at the Spectrum end is a humble 3.5mm jack plug, which should be inserted into the network socket. If you haven't got an Interface 1, don't despair — you can control the VL-5 via the Spectrum cassette port, using a small amplifier to step up the signal. Alter line 510 to read 'OUT 20,—("0"=A$(A)):NEXT A' if you don't intend using the Interface 1.

If your cassette recorder will amplify Spectrum sound effects, a wire from its earphone socket should give sufficient voltage (about 3 volts peak-to-peak) to drive the synthesiser. Alternatively, a small transistor amplifier (such as an intercom or baby-alarm) should work adequately if you turn it up loud and take the signal for the Casio from the loudspeaker.

SPECTRUM TO CASIO INTERFACE

COMMENT	LISTING
Print the instructions.	`300 PRINT " BAR CODE CREATOR" ' ', "By Sam Goodwin" ' ' ' " Enter the music in hexadecimal.Use'0408' as start of PITCH DATA or'0404' for LENGTH DATA." ' ' " This will be converted tobar-code data and output via the network interface." ' ' " Just connect to the ground and top right pin of any CASIO MS socket to transfer data to the musical instrument."`
Accept the music string.	`420 INPUT C$: LET C=LEN C$`
Work out the checksum for the music entered.	`430 LET F=3: LET A$="00": FOR B=1 TO C+3: LET D=CODE C$(B)—55+7 *(C$(B)<"A"): LET F=F+D`
Add the end-marker, taking its checksum into account.	`440 IF B=C THEN LET F=F—INT (F/16)*16: LET C$=C$+"F"+CHR$ (63+2 * (8*(F>7)—F))+CHR$ (48+8+ (F>7))`
Build the binary form of the music from C$, in A$.	`450 FOR E=B*4 TO B*4+3`
Force a change of state after every two bits.	`460 LET D=D/2: IF A$ (LEN A$)=A$ (LEN A$—1) THEN LET A$=A$+CHR$ (97-CODE A$ (LEN A$))`
Add a binary digit to A$.	`470 LET A$=A$+CHR$ (48+ABS (D-INT D-E/2+ INT (E/2)))`
Do each bit in turn.	`480 LET D=INT D: NEXT E: NEXT B`
Output the wave represented in A$ to the network.	`500 PAUSE 10: FOR A=1 TO LEN A$` `510 OUT 20, "0"=A$(A): NEXT A: STOP`
Draw the wave on the screen.	`520 FOR A=1 TO LEN A$: PLOT INVERSE 0;0,A*2: DRAW INVERSE "0"=A$(A);255,0: DRAW 0,1: DRAW INVERSE "0"=A$(A), —255,0: NEXT A`

SPECTRUM TO CASIO INTERCONNECTIONS



This is all the wiring you need to connect your Interface 1 to a Casio VL5. The socket on the Casio is a rather odd variety 4-pin DIN.

PRINTING BAR CODES

If your ZX printer is on good form, it's actually possible to print out bar codes. Amazingly enough, the resolution of the printer is usually good enough to read them as well — so long as you use good paper and avoid smudging the printout that is. Line 520 demonstrates this, although it only works for small tunes since it is difficult to imagine why anyone should want to print musical bar codes! Enter your music in the usual way and when the program STOPs after the music has been transmitted, type CONT (on the C key). The bar codes should appear on the screen. Type COPY to transfer them to the printer.

The one-line printing routine will only handle short bursts of music, since it simply prepares a single screen of

NOTE CODES

0	Rest
1	C natural
2	C sharp
3	D natural
4	D sharp
5	E natural
6	F natural
7	F sharp
8	G natural
9	G sharp
A	A natural
B	A sharp
C	B natural
D	Rhythm on
E	Unknown
F	End of bar

This table summarises the relationships between the digit and the note produced.

MUSIC ON SPEC

data. It should not be difficult to adapt the technique to produce longer bar codes. The routine deliberately uses the full width of the paper, so a succession of screens can be printed one after another. Another advantage of using the whole width is that it gives more chance of success with a smudged listing — if a pattern won't load at first, try moving the bar code reader across a different column of the printout. Producing bar codes with the ZX Printer can be a rather 'hit-or-miss' process, but perseverance pays off eventually!

SAFETY NOTES

The chance of damage to either synthesiser or computer is very slight when using this technique, but dire consequences could result if you are either very imaginative or very unlucky. To minimise the chance of problems, always use the synthesiser on battery power while it's connected to the Spectrum (this reduces the risk of damage caused by faulty mains wiring). Always double-check your connections between the two devices, and *never* connect them together while either is turned on. You can damage your Interface 1 by shorting the network output, so be especially careful that you've not accidentally connected the two wires together.

In our tests we've experienced no problems but, of course, *Your Spectrum* can't take responsibility if you blow up one of your 'toys' in the course of experimentation. At least you know that it's possible to interface the two machines — just spare a thought for us fiddlers and the projects that didn't work, as we sit in the midst of an expensive pile of broken robot-arms, ZX80s and central-heating controllers! YS



Well, I'll bless it if you think it'll help, but have you checked your disk drive?

NOTA BENE

When this article landed on the Editor's desk, cries of glee emitted from the would-be *Keith Emersons* who rushed down to *McDonald's Stores* of Oxford Street to borrow a VL-5. No problem — the device worked perfectly with the Spectrum, and within minutes the *YS* office looked like a scene out of *Fame*! However, although you can still probably pick up a VL-5 from your local electronics shop, Casio have ceased making this model. *But*, Casio do include other synthesisers in their range with a light pen port and, just to prove that the program works for these as well, *YS* tested the MT-70 model which you'll see illustrated within this article.

YS HARDWARE REVIEW:

YOUR FLEXIBLE FRIEND

Two disk arrangements, based on completely different ideas, have been available for use on the ZX Spectrum for some time now. Stephen Adams compares and contrasts the Viscount interface and drive from Spectrum Computer Stores and the FDI interface from Technology Research Ltd.

The object of buying a disk drive is to be able to speed up LOADing and SAVEing of data and programs — data being the most important, as files used for databases frequently have to be LOADed, MERGEd or altered and then SAVEd again.

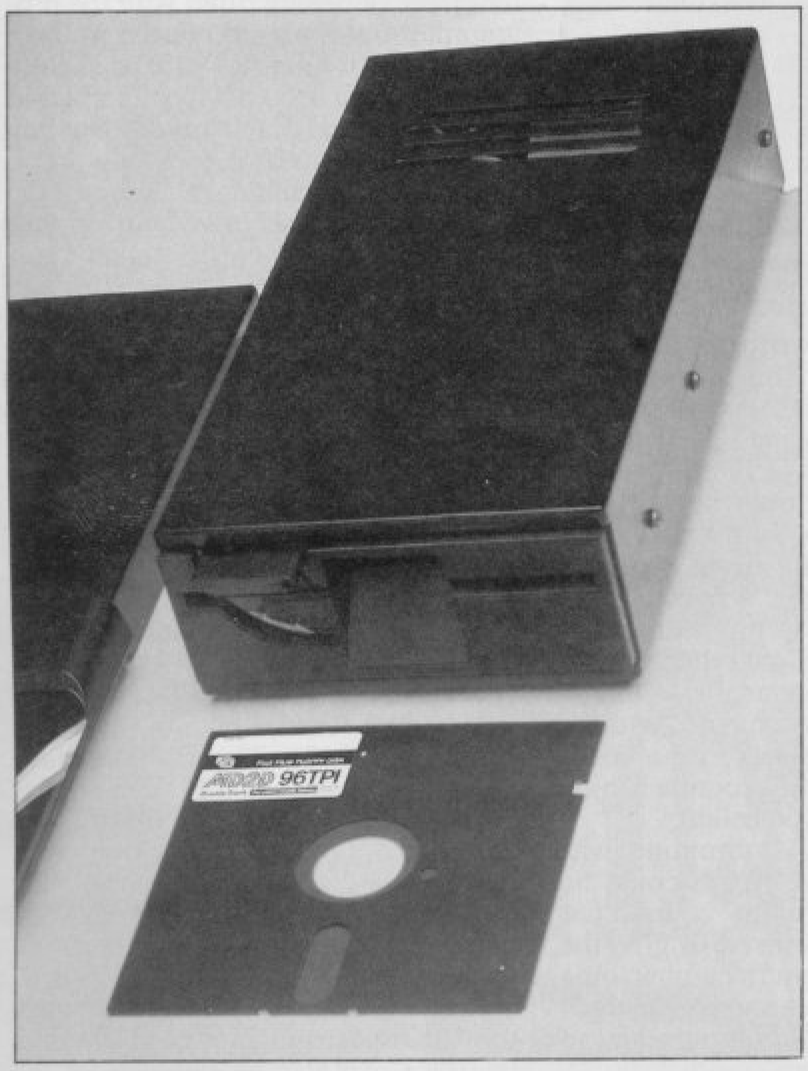
One advantage these disks have over tapes (including the ZX Microdrive) is that it's possible to go directly to an area (or record as it's called) of data and read it directly into the computer. This is called Random Access. Of course, on tape, the whole thing has to be gone through right up to the point where a particular record is found. And if it's not found then the tape has to be re-wound and the search begun all over again. This laborious process is known as Serial Access.

Another advantage is speed. The baud rate for LOADing data from disks is usually in the region of 128K baud as against the Spectrum's tape speed of 1500 baud, an increase of about ninety times! Double density disks and multiple drives can also raise the amount of data immediately available to the computer. The approximate data capacity of the systems examined here is about 100K minimum.

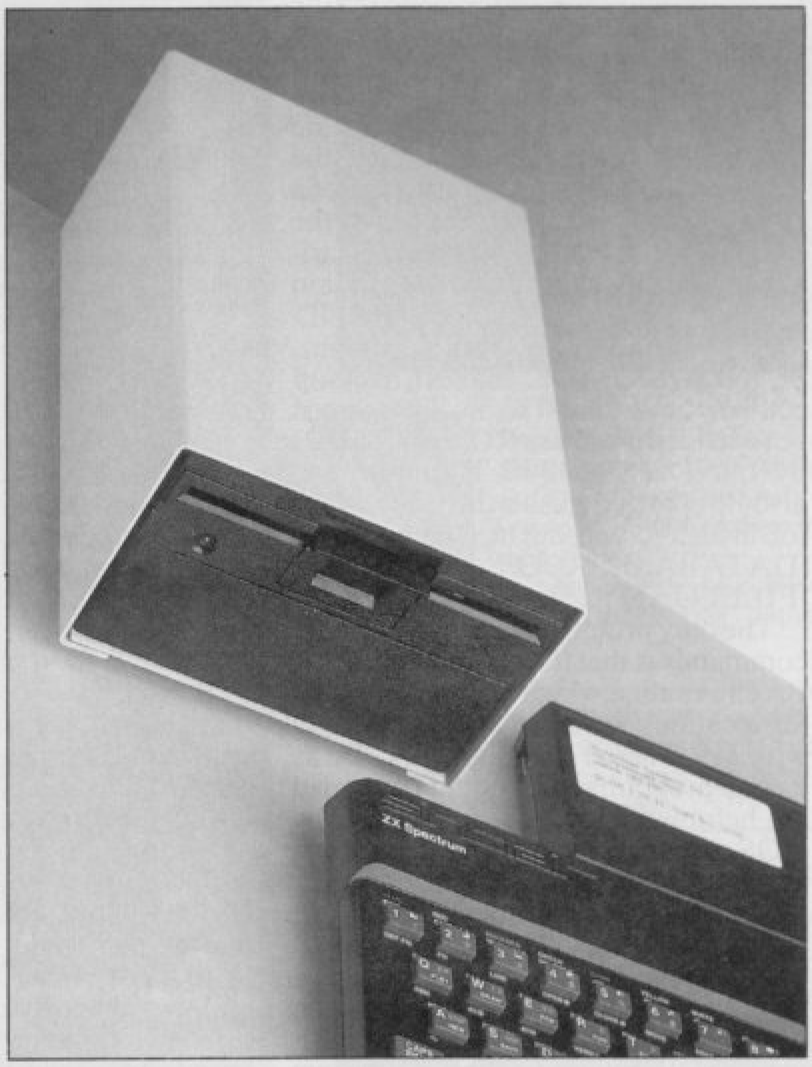
In both cases, connection to the ZX Spectrum takes place via the expansion port, but only the FDI provides you with a further expansion port at the back for attaching printers and the like. To use a printer with the Viscount, you'll need a special flexible cable with extra sockets that contain printed circuit cards. These are available from CPS (wholesale distributors for all Sinclair Research 'out of guarantee' spares) as well as Spectrum dealers.

Both systems use up memory space for the Disk Operating Systems (DOS) at the top of memory. The Viscount, because it's software controlled, uses up more memory (8K) than the FDI which normally only uses 1K. However, the FDI swallows up an extra 4K as a buffer for deleting files, and this needs to be allowed for when writing or modifying existing programs. And remember too that many machine code programs such as *Tasword* are position dependent in memory and you'll need a new version to work with disks.

Having inserted the interface in the back of the ZX Spectrum, powered up the disk drive (the FDI interface takes



The Viscount Interface plus drive.



The FDI Interface needs a BBC or similar self-powered drive.

YOUR FLEXIBLE FRIEND

BBC disk drives, so the choice is a wide one) and switched on, what do we get now? Well, on the Viscount we notice no difference at all except that there are now two copyright messages on the screen, and RAMTOP is 8K lower. With the FDI you are straight into the DOS and the system requests a password. This is 'TRL' (supposedly standing for 'Technology Research Ltd') for the disk supplied, but you can change it to anything you like. Only the correct password (which is recorded on the disk when it's formatted) will allow you access to the directory (a menu of the programs on the disk) and thus to the files.

Once through the password you may enter a command for either A or B disks (if you have a dual drive system) from the table listed at the end of this article. This command will then be executed by LOADing and RUNning the file named. You can return to Basic, but this NEWs any resident program.

Life is a little easier on the Viscount. You can LOAD a program from tape, enter a program or LOAD one from disk using the variables already inserted into the variables section by the DOS. All commands must be put into f$ in the form of name and the details of the file you wish to LOAD or SAVE. By file (as you might have already guessed) we mean a program, data array or machine code routine. The commands on the Viscount are easy to use, consisting of a call to a machine code routine in the DOS using instead of numbers, a variable which is a reasonable description of the command. For instance, 'RAND USR BL' LOADs a Basic program, 'RAND USR BACK' makes a back-up copy of the disk, and so on. These must be re-loaded if you use RUN or CLEAR by RAND USR 64000. The names can also be changed to suit the program — for instance, you might change BL to 'DATABASEPROG' and CL to 'FILELOAD'.

The only problem I found using these commands is that there's an error in the BACK routine which requires POKEing locations 63208 with 80 and 63209 with 249. These I eventually SAVEd as a code program which could be loaded up before using the command. The DOS may therefore be modified in both systems to eliminate errors.

I also had some trouble with the disk units themselves. They tended to overheat when I first got them, but this I cured by installing larger heatsinks on the power supply; there's been no trouble since. Problems on the TRL system centred mainly around the fact that I couldn't format my double-sided disks, despite trying several different sets of software from the suppliers. Eventually, I switched to a single-sided, single drive system which worked fine.

A word of warning, however. The interface socket, although quoted as BBC compatible is in fact upside down to the normal connector. Therefore, the notch on the plug *does not* go into the hole provided on the socket — at least not without causing damage to the interface. There's a listing of the pin connections in the manual, but no warning.

VIVE LA DIFFERENCE!

The FDI is designed to cope with machine code programs, mostly with the user allocating a buffer for the routines or data. Basic programs (using an '$' prefix) can be LOADed or MERGEd, but the variables (with a '#' prefix) must be SAVEd and LOADed separately. Any progam without a prefix is a machine code program which is LOADed and run from the location given. Basic programs may only be RUN on LOADing (no GO TOs). The commands PUT and GET allow you to SAVE and LOAD data in 512 byte machine code sectors into a buffer.

The commands in Basic must be POKEd into a workspace contained in the 1K area; but a USR routine must first be called to find out where. After POKEing the command in, a list of numbers within the command must also be POKEd into another series of locations. Then, the USR routine which looks at the line can be called.

Although a Basic subroutine was included on the disk, which could be MERGEd with the main program (so that strings could be used to give the commands), I still found it cumbersome and would hardly call it user friendly!

The FDI moves all programs and so on around to take up any spare space — which is very efficient. This means any new file is always added on the end of the list. The Viscount, however, only allows you to use complete tracks for files, which are 2816 bytes long. Thus if a program or data is only 20 bytes long it still takes up 2.816K.

Arrays can be stacked together in one file to give better use of the space and member numbers act like DIM numbers for the particular array you want.

THE VISCOUNT'S DOS COMMANDS.

Variable name	Command
ND	NEW the disk (format).
DIR	Print the directory.
BACK	Copy the contents of a disk on to another.
ZAP	Delete files from a disk.
BS	Basic SAVE.
BL	Basic LOAD.
CS	Code SAVE.
CL	Code LOAD.
AS	Array SAVE.
AL	Array LOAD.

A typical call to the Viscount's DOS could be: LET f$="GAME,(line number of autostart): LET ERROR = USR BS.

THE FDI'S DISK COMMANDS

COMMAND	OPERATION
DIR A:	Print the directory.
REN A: N Y	Rename file N to Y.
SAVE A:$FRED	SAVE file (no prefix denotes that it's a machine code file).
LOAD A:$FRED	LOAD file (as above).
ERA A:N	ERASE file from disk.
MERG A:$FRED	MERGE data or program.
GET A:N	Random read.
PUT A:N	Random write.
A:	Select drive A.
B:	Select drive B.
PASS	Password change.
INIT	Insert the directory.
LOCK	Protect disk.
BAS	Return to Basic (NEWs the program).

Commands must be POKE'd into a workspace and numbers calculated before using routine.

TOP OF THE FLOPS?

Both manuals are reasonable, given that they've been written with the programmer in mind and not the average user. The Viscount is perhaps slightly better in that it lists the error numbers at the back — the FDI manual leaves you to find out for yourself!

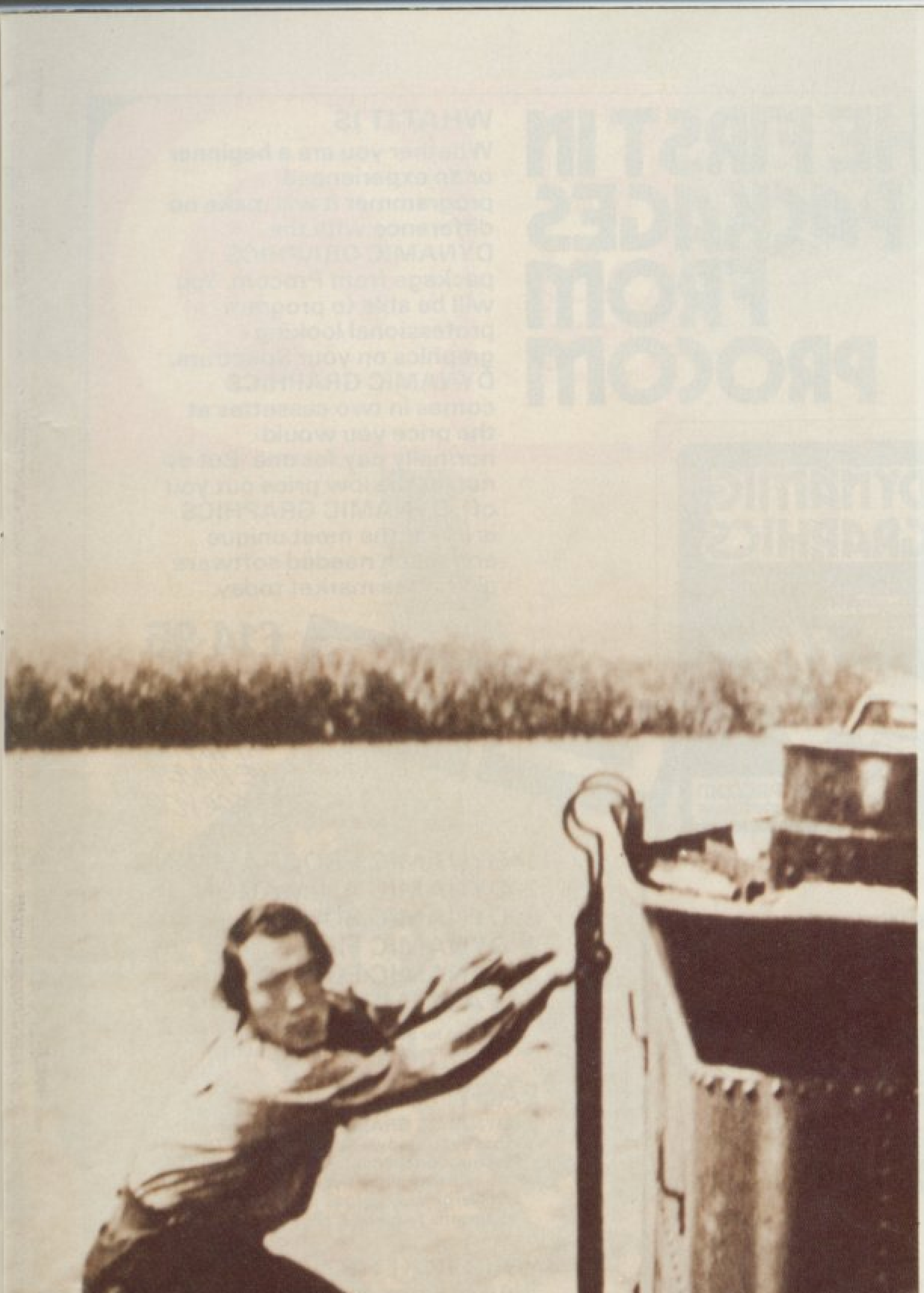
Taking everything into account, the Viscount would seem to be the better system. It not only costs less, but is far easier to use; it also includes a back-up program (very important when using disks), whereas it's an optional extra with the FDI.

DATA

The Viscount is only available at present through the Spectrum chain of dealers, but may also be stocked by other sources as it becomes more popular. The disk drive plus interface costs £245.

Technology Research Ltd is at 356 Westmount Road, London SE9 1NW (Staines 63547). The FDI interface costs £99 and needs a BBC or similar disk drive (self-powered); TRL's equivalent to the Viscount disk costs £218.50 and a connecting cable adds perhaps a further £15. The copying program (to back up the disks) costs an extra £7.

Both systems reviewed here require the 48K Spectrum. However, the FDI interface can also be supplied for the 16K machine. YS



Still of Buster Keaton from The General © Raymond Rohauer 1984

Try pulling a stunt like this on your Spectrum

Leap on a moving train ... jump from carriage to carriage ... duck under fast-approaching bridges.

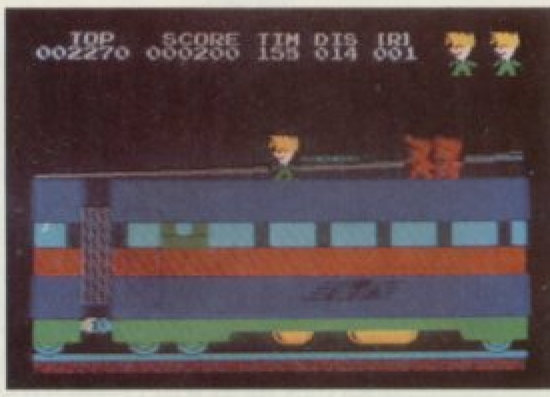
But before you do *anything*, make sure it's with Sinclair's new action-packed game – 'Stop the Express'.

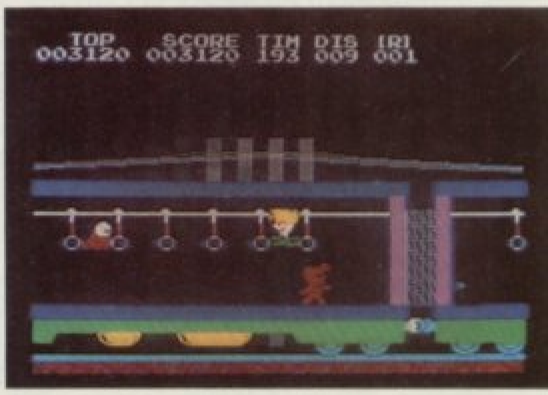
That way, you can try dramatic feats like Buster Keaton's – without risking your neck!

'Stop the Express' is one of five exciting new games from Sinclair. Its graphics are superbly fast and sharp. The action is non-stop. And the challenge is a very tough one.

On top of a racing express, you're chased by knife-throwing bandits. And inside the express, there's even more trouble.

Will you ever get to the front and halt the train? Or will you be well and truly bumped off?





All five new Sinclair titles – Stop the Express, Bubble Buster, Zipper Flipper, Eric and the Floaters, and Driller Tanks – are for a 48K Spectrum. You'll find them in the shops – today.

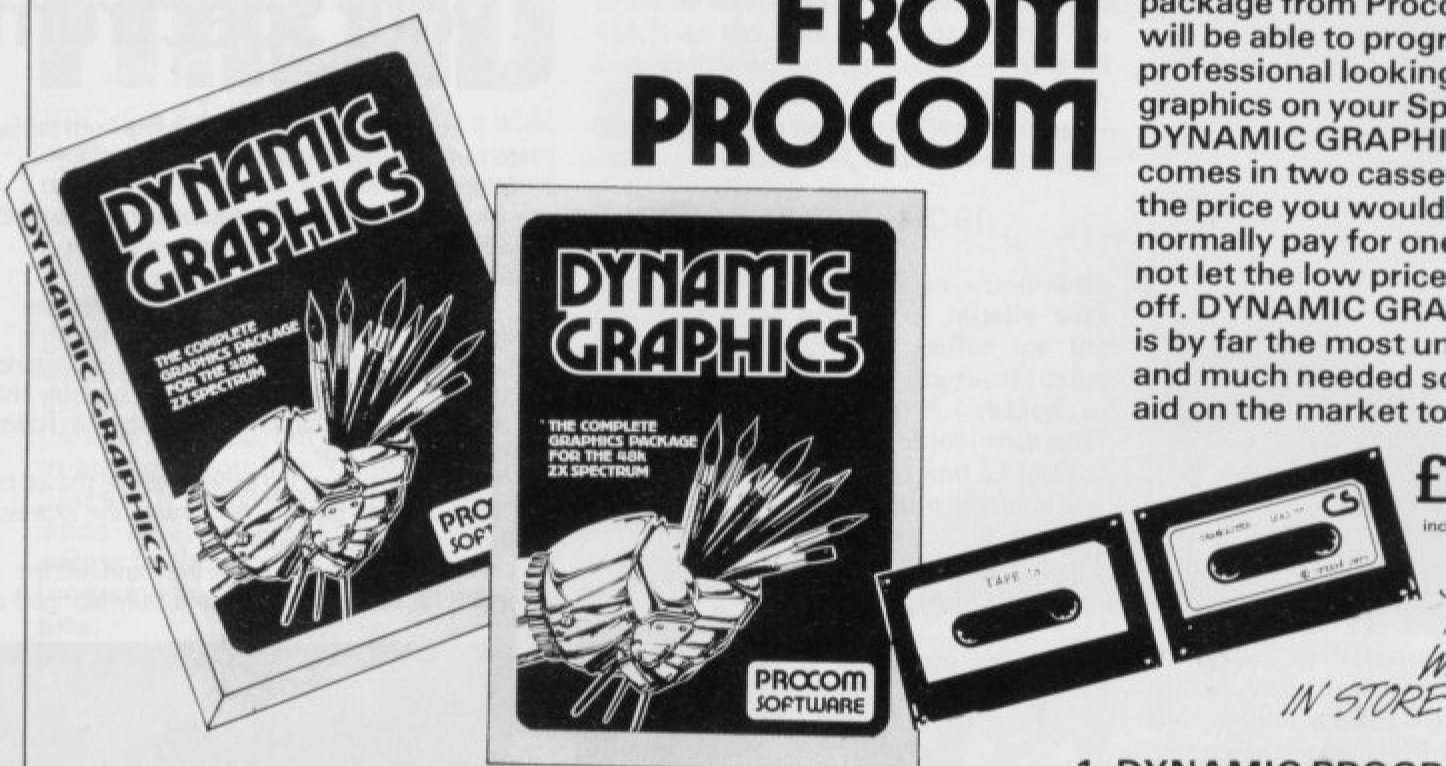
At only £5.95 each, they're destined to be big stars on the small-screen!

Selected Sinclair software lines are available from W.H. Smith computer stores, larger branches of Boots, John Menzies, Greens and most other software stockists nationwide.

Sinclair Research Ltd, Camberley (0276) 685311.

sinclair

THE FIRST IN DYNAMIC PACKAGES FROM PROCOM



WHAT IT IS

Whether you are a beginner or an experienced programmer it will make no difference with the DYNAMIC GRAPHICS package from Procom. You will be able to program professional looking graphics on your Spectrum. DYNAMIC GRAPHICS comes in two cassettes at the price you would normally pay for one. But do not let the low price put you off. DYNAMIC GRAPHICS is by far the most unique and much needed software aid on the market today.

£14·95
inc. VAT and P&P.

JUST LOOK AT WHAT WE HAVE IN STORE FOR YOU!

1 DYNAMIC PROGRAMMING
2 DYNAMIC ANIMATION
3 DYNAMIC SOUNDS
4 DYNAMIC FILING
5 DYNAMIC EDITOR & ASSEMBLER

Package for the ZX Spectrum now! (Also for the BBC, Electron and CBM 64)

TAPE 1

THIS IS A CHARACTER DESIGNER WITH A DIFFERENCE * FULL SCREEN ANIMATION OF SPRITES * 27 FUNCTIONS INCLUDING SPRITE HANDLING 96 CHARACTERS WHICH CAN BE SAVED ON TAPE TO USE IN YOUR OWN PROGRAMS * EASY METHODS OF INCLUDING CHARACTER DESIGNS IN YOUR OWN BASIC PROGRAMS * NO KNOWLEDGE OF MACHINE CODE NEEDED * SCREEN SAVING * SPRITES MAY BE TREATED AS SUPER LARGE CHARACTERS (4 x 4). THE LIST IS ENDLESS. YOU WILL BE ABLE TO DESIGN LITERALLY THOUSANDS OF DIFFERENT CHARACTERS FROM PAC MEN TO SPACE MEN OR SPACE SHIPS TO BATTLESHIPS.

TAPE 2

THIS IS THE ULTIMATE UTILITY FOR COMPOSING HI-RES GRAPHICS ON YOUR SPECTRUM. AS WELL AS ESSENTIAL HAND DRAW CURSOR CONTROLS (OR JOYSTICK) THERE ARE MANY ADVANCED FEATURES, SUCH AS FAST FILL AND RESCALE. THERE ARE IN ALL OVER 50 COMMANDS. A CONSTANTLY UPDATED STATUS DISPLAY. GENERATED PROMPTS ARE FULLY DISPLAYED. NOW YOU CAN DESIGN SCREENS AS GOOD AS THE SOFTWARE COMPANIES.

EASY TO USE

DYNAMIC GRAPHICS is the one that everybody will understand. The instruction manual is simplicity itself. No more strained eyes — the manual has large 10 x 8 pages with plenty of diagrams to make it easy for you.

YOUR TICKET TO RICHES?

Use DYNAMIC GRAPHICS to make up your own program, and if it's good we will market it for you and maybe make you rich.

DYNAMIC GRAPHICS

PROCOM SOFTWARE

309 HIGH ROAD
LOUGHTON
ESSEX
Tel: (01) 508 1216

HOW TO ORDER

Telephone orders
(01) 508 1216

Credit Card Holders (Access/Barclaycard only) can purchase by telephone. Please give card no., name, address and items required, and your order will be despatched the same day.

BY POST

Simply write your requirements in the coupon provided, enclose your cheque/P.O. or use your Access or Barclaycard. Please make cheque payable to; PROCOM SOFTWARE and post to address below. Allow 7 days for delivery.

IN PERSON

Visit our premises, at the address below for a great selection — we will be delighted to see you.

Post to: PROCOM SOFTWARE, 309 HIGH ROAD, LOUGHTON, ESSEX.

Please send me the following items:

☐ DYNAMIC GRAPHICS ☐ DYNAMIC PROGRAMMING ☐ DYNAMIC ANIMATION
☐ DYNAMIC SOUNDS ☐ DYNAMIC FILING ☐ DYNAMIC EDITOR & ASSEMBLER

(Please tick items)

I enclose my cheque/P.O. for £ ____________

I prefer to pay with my ACCESS/BARCLAYCARD
(Delete whichever not applicable)

Card No. ____________
Signature ____________
Name ____________
Address ____________

Tel: (Day) ____________ Tel: (Eve) ____________

Credit cards valid if signed by card holder.
Address above must be the same as card holder.

YS284

EMS

QL USER



WHY IS THIS MAN SMILING?

Full details inside...

Steering To Success:

The sordid practice of pre-announcement has again gripped our hearts, and Sinclair are once more laughing all the way to Fort Knox. But to keep the interest ticking over during The Big Wait our ever-smiling micro Guru stuffed a company Mercedes full of eager journalists and whisked them over the Cambridge for a day's play with a QL. Quentin Lowe tells the tale.



The QL Roadtest



If nothing else, the Cambridge soiree succeeded in wiping a few cynical smiles off a few disbelieving faces. Yes, there are mistakes and disappointments; yes, there are delays and problems that a two-year old could have predicted; but it's one hell of a machine all the same and Sinclair Research knows that, in the end, it will delight everyone who's had the good grace to put their money up front.

Before charging into the review, I must first make all the usual disclaimers. The report is based on a one-day play on an early machine. Details may be different when the machines finally appear (in fact the QL has already changed a bit since the last time I saw one) and nothing that you read can be taken down and used as evidence...

Surprisingly, the man from Sinclair said how disgusted they all were that anyone could have written and be selling a book about the QL. But books about mythical computers are only about a fiver, whereas this particular mythical computer is going to cost some four-hundred pounds. Still, Sinclair Research did at least prove that the QL was slightly more than mythical; here's what it's like to ride a Unicorn...

THE BODYWORK

The machine is delightful to behold and use. Its small size makes it a comfortable desktop companion and the only possible moan is that the three plastic feet (which tip the machine to a suitable typing angle) have a habit of falling off.

Early QLs will have an additional 16K of EPROM hanging out the back — an unforeseen extra that's affectionately been nicknamed the 'Kludge'. The Kludge is really a 32K EPROM or ROM cartridge and its current nasty looks should have changed somewhat by the time it reaches the customers. It contains the bits of Basic and QDOS that either weren't ready in time or couldn't be made to fit in the original 32K ROM space. Being an indispensible cartridge, it must have a postponing effect on the QL cartridge software market.

In a way, Sinclair's approach is commendable; with the Kludge, you do get a working QL as opposed to yet another apologetic letter. The final QL ROMs will probably be around some four months after Sinclair has frozen the code in them — and the company also has to figure a way of getting 48K of ROM inside the machine in a relatively short space of time. Strange then that it seems to be moving so slowly to finish the code in the ROMs. You also hear rumours of this or that keyword being added/deleted/changed and, despite missed delivery dates, Sinclair Research reckons its got the time for its programmers to do some new and fun things with the extra space available. Oh well, so you get turtle graphics commands too.

The keyboard is a bit strange at first. There's a copious layout (rumoured to be close to a DEC-like standard — if that were at all important) and every key you could possibly want is provided, including obscure characters such as {}, [,], ¦ and \ that come in useful in languages such as C. What's more, the keys feel good and with a bit of practice you can type reasonably quickly. Best of all, the QL never

drops a keypress. It's got a proper type head system like the BBC micro, especially useful when you're waiting for the Microdrives.

However, you will notice that there is no delete or backspace key, a curious omission considering its importance. Sinclair has a complete editing key suite between the arrow, Control and Alt keys which will be available in most applications. This includes Control/Left-arrow for backspace. I hate to say it but after five hours, I was quite happy with it. Any Shifted Zs you see in this story are purely coincidental!

DRIVEABILITY

As soon as you've switched on (and are still cursing the lack of an on/off switch) the QL asks if you're using a TV or Monitor. Pressing F1 or F2 sets the appropriate display mode and the QL then tries to boot the Microdrive cartridge in drive 1 (or as we QLers call in MDV1). The QL searches MDV1 for a Basic program called BOOT and then runs it. This can then load whatever other programs you like, allowing you to create turnkey systems or have the QL boot up just how you like it. If it can't find boot, the system drops neatly into Basic.

The choice of TV or monitor is only for aesthetics. Some TVs won't be able to handle the QL's 80 column mode and will have to be operated in 40 columns. Sinclair, a company which has no televisions in its office, assures us that the television display is as good as you'll get. Certainly on a monitor it's bright, sharp and rock-steady — even if one of the QLs we've seen lost the left-hand 50 pixels of the display off the edge of the screen.

Amid the growing pile of QL blurb, you tend to lose sight of the display's performance. It's a pure bit-mapped display with its own 32K of memory, operating in four colours at 512x200 or eight colours (which, with a bit of software hocus-pocus, provide 256 stipples) in 256x200 mode. Sinclair has built some pretty clever screen handling software into QDOS and no doubt there'll be lots more to come.

Needless to say, the QL's windowing facility is a joke. The windows are not proper windows in the Apple, Digital Research or Microsoft sense, but more a clever screen handling gimmick. A window is effectively a set of four margins, a top left, PAPER and INK colour, cursor position and so on. It's associated with a particular I/O stream so that screen handling commands can be directed specifically to it.

So although it may look like it's a window, it has no contents as such — it can't be opened and closed with stuff inside. There are no window priorities — print to a window behind another one and the new text shows through, destroying whatever was in that place on the top window. Clear a window on top of another one and you destroy its contents and so on. There are PAN and SCROLL commands but these just smooth scroll the area inside the window in any of four directions, losing any data that's scrolled out of bounds.

So, with a great deal of care and attention, it is possible to produce some stunning effects. But the windows are just a trick piece of screen handling software. Similar things could easily be achieved on any other microcomputer, including the Spectrum. You may have guessed by now that QL screen handling is very software based. Character generation certainly is, and the eight different sizes available with CSIZE are just the same character definitions trotted out on to the screen in different ways.

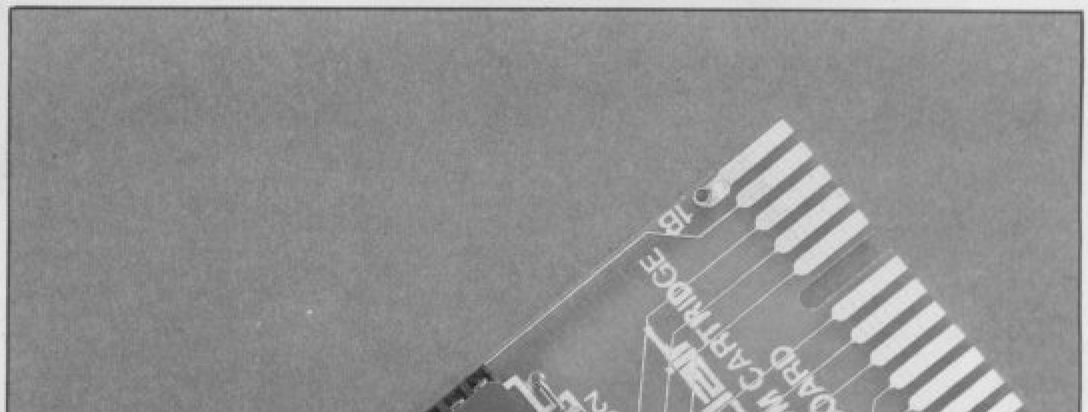
Following the shock of the impressive display, the next surprise is the two Microdrives. Despite the now wearisome round of moans about Microdrives, the two souped-up versions in the QL are a major asset. Having any kind of viable mass storage device in a home computer transforms the system. No-one is going to suggest that Microdrives are fast but they are not as slow as people might think.

The QL uses a slave processor to control the drives (it even verifies tape-writes in the background while you get on with something else) and a proper RAM-based directory — unlike with the Spectrum Microdrives. Occasionally you find yourself in exactly the right place and the drives load ten lines of Basic before you've looked up at the screen. Occasionally, they get fussy and you sit there for hours and hours.

THE KLUDGE

Early QLs will have a 16K EPROM kludge containing the remainder of QDOS and SuperBasic that didn't fit in the original 32K ROM space. Unfortunately this rather monopolises the cartridge port. On the bright side, however, the extra space did give Sinclair Research room to add some turtle graphics commands.

Rumours that the kludge is merely aesthetic and designed to make the QL look and feel like a true Sinclair Research device are, of course, nonsense!



COVER PHOTOGRAPHY TONY SLEEP/ HARDWARE PHOTOGRAPHY IAN McKINNELL

The media itself is nothing short of fun. The little cartridges can be carried round in pockets, put in Jiffy bags and so on. Despite stern warnings, you can do nasty things like pulling them out while running without wrecking them. This unfortunately became a necessity on a number of occasions while using the Psion packages, but that's a software problem that should be curable.

QDOS, incidentally, is very good for friends who swap cartridges around ad-hoc. When you format a tape, it gets named plus it gets a random number written to it. Even if you switch one cartridge for an identically named one, QDOS can detect it and object, thus protecting you from the consequences of the RAM-based directory getting out of sync with the actual tape in use. This process, while quite neat, does unfortunately slow down access.

All in all, it's hard to knock anything that's such good value for money. The drive system is probably the QL's best feature — it could be a Z80 with 32K and the drives would put it above its rivals. The only real way to complain about the Microdrives is the sobering thought that when you can finally buy the QLs in Smiths, there'll probably be twin 5¼ inch floppy machines for around the same price.

The more disappointing aspects of QL hardware are to be found lurking round the back. The omission of a Centronics port ranks in order of stupidity only second to the lack of a delete key. Instead we face a strange obsession with poorly implemented serial ports. First point — daft sockets. The QL uses telephone jacks which, although very trendy, aesthetic and (let's face it) secure, aren't easy to come by and are more than difficult to wire up yourself. Sinclair's wonderful free-gift for its 'sponsors' who've already ordered is, of course, one of these rare RS232 leads, valued at £14.95. If you're going to use such daft connections, the leads ought to come in the box, value £0!

But worse, the two serial ports have to be set at the same baud rate and that rate applies to both transmitted and received data. Think nothing of it until you try to access Micronet (1200 receive with 75 transmit) or drive both a Modem and a Printer. Who knows, maybe some flash device driver can squeeze other rates out of the system.

The network is quite openly not much more sophisticated than the ZX Interface 1 network. Sinclair admits to, as yet, having problems with Spectrums on a net to QLs (who wouldn't?) but is able to demonstrate two QLs talking to each other. For the most part, it's still a question of two people stopping what they are doing and negotiating every step of the data transfer (using the NET command and the NETI (net input) and NETO (net output) drivers in QDOS). For the inexperienced, this can take more time than passing a Microdrive cartridge across by hand. However, with QDOS and the 68008, a sensible network should be possible. It remains to be seen whether the built-in system is up to it.

STEERING IT

Onto the software and, like all things Sinclair, SuperBasic plods along as if all it had to do in life was waste away a warm summer evening by the river. It's a glamour Basic that does have some really juicy features, but what people ought to realise is that it's speed that counts. Basics are habitually used for stuff like games and it's their ability to crunch code quickly that makes them good, rather than oversized ranges for floating point variables and so on.

"SuperBasic plods along as if all it had to do in life was waste away a warm summer evening by the river."

SuperBasic also has a lot of the Cambridge academic in it. Long, cumbersome commands and short, terse error messages. The latter seems to come from using a predefined set of rather vague messages provided by QDOS. Quite why this is going on seems too odd to be worth investigating.

SuperBasic has all the makings of a grown-up language. Multi-line functions, procedures with local variables, long variable names and formatted listings with words like REM expanded to REMark for you. SuperBasic does away with the humble NEXT command — instead you can use END FOR. Using NEXT allows you to jump out of FOR...NEXT loops whenever you feel that Sinclair's claim of Superbasic making GOTO and GOSUB redundant is a little crass. To be fair, however, if you use FOR followed by a series of statements on a single line, you don't need the NEXT (or END FOR) at the end. Most elegant.

The other major nicety is that procedures can be executed outside of programs simply by typing their names. Given that these procedures can have parameters, you can easily add your own commands to SuperBasic to make program development that much more simple. For the most part SuperBasic appears to be a very sensible implementation, although missing some of the essentials — a line number trace being one of the more sore points.

It's a mature language, but one which appears to assume a certain familiarity and skill with Basic. Normally that's not the objective at all. To my mind it's for beginners and should therefore provide a fast and friendly environment that makes it easy to learn and use.

BACKSEAT DRIVING

The four bundled applications were looking a bit sorry for themselves on the day, probably because Sinclair Research gave the impression that there were more releases and versions of the packages than working QLs — and because they are still some way from being presentable. However, running versions 0.99C, did give a fair impression as to what they are about.

The packages are real business programs and have no rival on any other similarly-priced machine. But they are not as slick as they first sound. The overall feeling was one of slowness, particularly with the word processor, *QL Quill*, which relies heavily on using MDV2 as a scratchpad. *QL Abacus*, the spreadsheet, also seemed more lethargic than you'd expect.

The packages now run under QDOS and are loaded from Basic BOOT programs. One thing that's been going on is that they've been steadily adapted to use more and more of the ever-growing QDOS. In places this may speed the system up (for example, fast area fills in graphics) or it may improve performance (QDOS supports 256 stipples based on the eight available colours). Hopefully, having to go through QDOS will not have a slowing effect.

There's no facility for linking the packages directly but they can exchange data in a predefined import/

export format. Going from the database to the wordprocessor produced a document with the database information plonked inside in an ordinary looking 'comma and quote' format. So it should be possible to access and create import/export files from Basic. However, the time taken to export a file, load another application and import the file is a considerable deterrent.

What's more worrying is the user-image of the programs. They look quite fabulous and exciting in the screen shots and over someone else's shoulder. But try sitting down and doing a job and you find a shockingly large amount of typing needs to be done and there are some fairly curious and fiddly key sequences.

You can get used to curious user-interfaces, but you can't cope with insecurity. Almost any filing operation causes the QL to hang, the ESCape key being ignored. If something is wrong with the cartridges in use the system hangs indefinitely, all the while re-trying the same operation. There are also a number of dangerous operations which should be checked before any move is made. Tell *Archive* to Quit and it resets the machine, wiping your current data, programs and screen formats, all without asking any questions.

No doubt much of this could (and probably will) be sorted out before machines start being shipped in any quantity. But the overall suspicion is that the packages perhaps try to do more than is necessary and maybe too have suffered in the rush. It's hard to believe they put a strain on the machine but it is easy to wonder whether someone has been a bit over-ambitious.

Many people have ordered QLs solely for the four packages and it's certainly not unreasonable because you get the Perfect suite. However, it's nowhere near as good as gritting your teeth and buying one of those ageing Osborne things on the grounds that you get *SuperCalc* and *WordStar*.

TRANSMISSION

QDOS, the QL's ROM-based operating system, does exist and by all accounts is an impressive piece of work. It wasn't really possible to tell how good it is because you can't use it directly and there's as yet no access to some of its more powerful features — a touch of the 'honest the really powerful features are there; you just can't use them' department! Neither is there any documentation; so much of QDOS remains a mystery and these details are perforce more sketchy than they should be.

QDOS is a proper operating system, despite being stuck in ROM rather than loaded from Microdrive on power up. Elsewhere the practice of putting serious operating systems in ROM has almost died out, since it makes it quite hard to cure bugs and update users with new versions. That forgiven, Sinclair appears to have produced a very mature system.

QDOS has been likened to Unix simply because it's 68000-based and uses lots of Unix style syntax. In reality it's nearer to something like MS DOS 2 with the promised bonus of multi-tasking. QDOS provides total hardware support (graphics, Microdrives, network and so on), multiple Input/Output streams with installable device drivers, multiple programs in memory with pipes and simple multi-tasking — although these latter features can't be demonstrated.

"Despite one or two harsh words, it's hard to deny that using the QL has made me want to own one."

At first glance, everything is beautifully implemented. All I/O devices have names (see table) and you OPEN and CLOSE streams to them, PRINT and INPUT to and from them. Some support other appropriate commands such as CLS and INK or LOAD and SAVE.

QDOS DEVICES

MDV1 MDV2	The Microdrives
CON	A window opened as a console (that's both input and output from the window)
SCR	A window opened as a screen (for output only)
NET	The Network port
SER1 SER2	The two serial ports

When you open a channel, you can pass an arbitary string of characters excluding commas, to the particular device driver software. This can be used as parameters for the driver so, for example, you specify a filename when you use MDV1 or MDV2. A command to OPEN a CONsole in a screen window includes a rather ungainly spec for where the window goes and so on. Sinclair has already let the syntax for such parameter lists get out of control and inconsistant.

The beauty of the system is that any new device can become part of the machine at will. A Centronics interface add-on will probably become a CEN device. To print to it, you would OPEN CEN with a parameter list that perhaps tells it what sort of printer it's using and whether a screen dump is required or not. There's room for 16K of code for each device, allowing for some pretty sophisticated additions to the system. Obviously, it makes it easy to bring in floppy and hard disk add-ons. QDOS recognises that devices are either character based (such as printers, keyboards, networks and so on) or directory based (such as Microdrives and hard disks).

Once the calls to QDOS are available, there's plenty of room for improving the system; a RAM-disk driver, perhaps, for half-megabyte QLs? Or a fancy font generator for the screen. With all this so nice, it's surprising that the company has made some really bad decisions with the rest of the operating system.

Files are kept absolutely simple — all are effectively just collections of ASCII bytes and QDOS will happily dump any file on the screen or down the network using the Basic COPY command. This is not such a bad idea — as a consequence, you can do little things like create programs for the *Archive* database with a Basic program or import from the spreadsheet into Basic and so on. SuperBasic programs are saved as straight ASCII so a straight file editor (not *Quill*) would provide a global editing facility.

QDOS is fine on the standard 128K twin Microdrive home computer, but the moment anyone puts a hard disk on the system, Sinclair Research will feel really stupid about the way it's done the DOS. There are almost no file types, no partitioning system and so on. This is fine if you have two drives with about ten files on each, but it makes a hard disk system unmanageable.

None of QDOS's great facilities — such as pipes and multi-tasking — are available from the keyboard through Basic. It's nothing short of lunacy to have the gear all nicely debugged and sitting there in the ROM, but to allow no access to it short of coding up your own machine code once the documentation is available.

It's a serious operating system,

so why does it have to hide behind Basic? In reality it needs a command line system like CP/M or MS DOS before it can be used properly. And for that matter, a cartridge of utilities — a text editor, for instance — would be very helpful. There's a chance that all this may happen, but not for some considerable time.

Even so, despite one or two harsh words, it's hard to deny that using the QL has made me want to own one. The machine, when it happens, is likely to have the same sort of impact and long-lasting appeal as the Apple II, IBM PC and Sinclair Spectrum. The system feels grown-up and organised and the hardware feels professional enough for serious work both in development and actual gainful employment.

For hackers, it's a dream come true — a serious computer that's got great graphics (don't mention the sound), a meaty Basic, fast loading and saving and a trendy processor. The bog-standard new user who bought it just as a desk-top tool will, for the money, see it in the same light. Shame that — as dreams go, this one is such a long time coming about.

SUPERBASICALLY . . .

SuperBasic boots with three windows open — the black one is for SuperBasic commands and error messages, the white one is for SuperBasic LISTings and the red area is for 'input to' and 'output from' the running SuperBasic program. This is automatically set up by two console channels (#0 and #1 — the red and black windows) and one screen channel (#2).

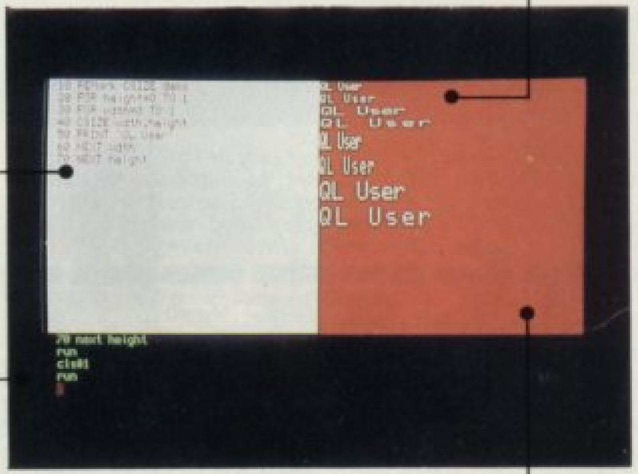
The three window system is designed to provide an easy environment for developing programs. However, the novelty quickly wears off as you discover some rather awkward problems. If you LIST a program twice, it appears in the white window with no line space between the two copies of the program.

More importantly, there is no command to change the defaults for these windows. To make the QL look like a normal Basic computer, you have to turn off Windows #1 and #2 and use explicit commands such as LIST#0 and PRINT#0. The problem of the default channels also manifests itself in other areas. All error messages generated by QDOS (and hence SuperBasic) are sent to channel #0. Close it (as the Psion packages do while loading) and you can't get any error messages!

To conclude — although SuperBasic may have some immediate visual appeal it is obvious that further research is required before it will be comfortable to use.

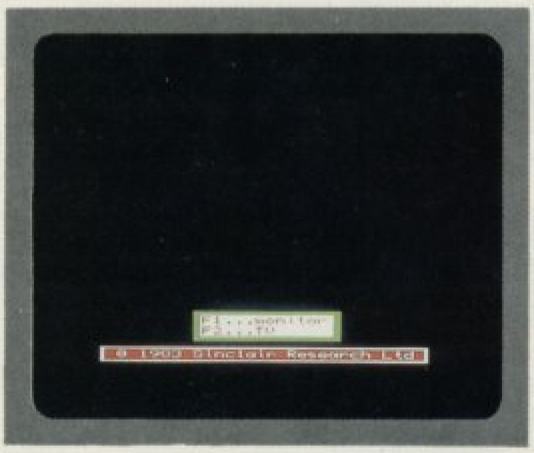
The SuperBasic listing builds in Window #2 as you enter it.

The CSIZE command produces two character heights and four character widths in software.



Window #0 is for commands and error messages.

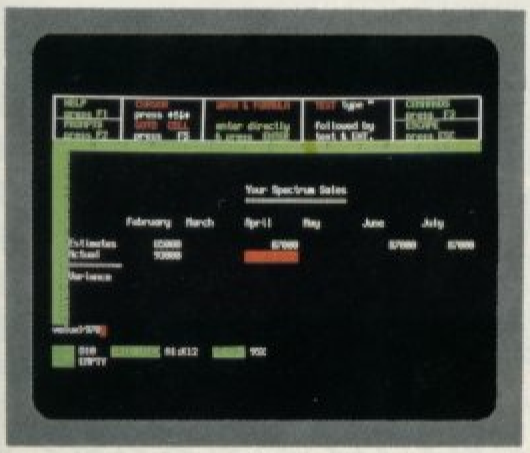
Output from the SuperBasic program appears in Window #1.



On power-up or reset, the QL needs to be told whether it's using a monitor or TV.



A Psion loading screen . . . get used to it, you're going to see a lot of it!



QL Abacus in action, even if it does run slightly slower than you'd expect.

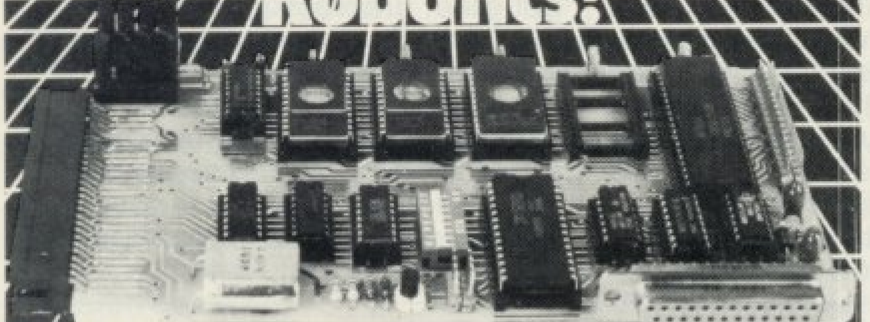


QL Quill is dramatic to watch but frequently very slow in operation.



A careful sequence of window commands moves around SuperBasic's default arrangement.

Robotics!



The 'FORTH' dimension for the Spectrum.

Now your 16k and 48k Spectrum get 12k of fig-FORTH from a new cartridge that's perfect for robotics.

This piece of genius is the creation of David Husband and it's the only ROM cartridge of its kind available.

It has RS232 and Parallel ports that not only facilitate remote control but are usable from FORTH or BASIC with the parallel also allowing a Centronics printer to be driven.

Due to an Interrupt driven 'Break' key the machine cannot be 'hung-up' and a number of routines and FORTH words are Vectored allowing reconfiguration.

Later in the year a software upgrade will be available which will permit multi-tasking.

Order the Spectrum FORTH I/O Cartridge £59+VAT using the coupon adding £5.75 p&p & insurance (£10 for Europe, £15 outside) or if you want more detailed information, tick that box instead. SUBJECT TO AVAILABILITY

Please send me Spectrum Forth I/O Cartridge. £59+VAT.
Tick whether for 16k [ ] or 48k Spectrum [ ]
Cheques to Skywave Software Readers' A/C (or enter Visa No.) Please send me more information.
[ ] Multi-Forth 83 [ ] ZX81-Forth ROM

Name ____
Address ____
____ Post code ____

Send to Skywave Software, 73 Curzon Road, Bournemouth, BH1 4PW, Dorset, England. Tel: (0202) 302385

Skywave SOFTWARE

SPECTRUM FORTH I/O CARTRIDGE

OLYMPICS 84

48k ZX Spectrum

To celebrate this summer's Olympic Games in Los Angeles, STORM SOFTWARE offers you an unique "two program" package. Crammed full of facts and figures – OLYMPICS '84 – gives great flexibility in looking at results.

The first program details the medal winners, by country, in every current Olympic event since 1896. The second program enables you to record the name and nationality of each medal winner in Los Angeles, this information can be analysed and compared with the full result in the previous two Olympic Games.

OLYMPICS '84 offers many extra features:

* Bar-charts to compare performances over the years, in any event, including Men versus Women.
* Medal tables, for any Year, Sport or Event.
* Ability to find a country's best sports and events.

OLYMPICS '84 package includes an amusing review of the Summer Olympic Games, from Athens 1896 to a preview of Los Angeles 1984.



This attractive package includes:

* **TWO programs**
* **24pp History of the Olympic Games, with original sketches**
* **Every Summer Olympics since 1896**
* **A MUST for all sports enthusiasts**

Available from all good computer shops. In the event of difficulties, order direct at £14.95+50p P&P (inc. VAT). Dealer enquiries welcome, contact: Amanda Ralph on 0935 813528.

Storm Software
Winchester House
Sherborne, Dorset DT9 4DL

STACK 100 LIGHTPEN - £28.75 *GIVES YOUR COMPUTER EYES!*

SLR
(STACK LIGHT RIFLE)



JUST TWO OF THE EXCITING PRODUCTS IN THE STACK 100 RANGE

Available for the CBM 64, VIC-20 and 48K Sinclair Spectrum, this quality rifle comes complete with three exciting games and connects to your computer with 12 feet of cable. ***The SLR puts you in a different league.***

£29.95

THE STACK 100 RANGE

CBM 64 Accessories

Cartridges:-
HELP - over 20 extra commands, disassembler and machine code monitor, DOS **£28.75**
SUPERHELP - as 'HELP' but with a comprehensive 2 pass assembler **£40.25**
ARROW - loads and saves a 32K program faster than a 1541 disk drive (use with 1530/C2N cassette deck) **£33.35**
ARROW PLUS - as 'ARROW' but with a comprehensive 6502 assembler **£44.85**
4-SLOT MOTHERBOARD - (switched) **£33.35**
and a full range of printer interfaces.

Please send me a Free brochure, price list and the address of my nearest stockist.
Name ....
Address ....
....
E. & O.E. Y.S.

STACK 100
CUSTOMER INFORMATION CENTRE
290-298 Derby Road, Bootle, Liverpool L20 8LN
Trade Enquiries: 051-933 5511 ask for 'Trade Sales

All prices are inclusive of VAT and delivery.

ADDING ZIP 2!

Moving on from the specification of his ZIP compiler,* Simon Goodwin's *back this month with the complete listing of the program along with explanatory notes of how it was written.

The Basic program for *ZIP* is listed in this article (Figure 1). It occupies about 16K including variables and REMarks, so it won't fit into a 16K Spectrum (which only allows you about 9K for Basic programs). On a 48K machine, however, you're left with 13K for the program you wish to compile and about 10½K for the compiled machine code.

Last month, we listed the instructions which the compiler will recognise, and explained the way *ZIP* uses memory. This month we'll show how the main program was developed. The whole compiler is written in Basic, yet it generates programs written entirely in machine code. Last issue we listed the 'library' of machine code building blocks. Figure 1 is the program that selects combinations of those blocks which work the same way as the Basic program — except much more quickly! For example, a Basic program to POKE patterns onto the screen will run more than 400 times faster once it has been compiled!

THIS WAY UP

The compiler program was developed using a technique called 'top-down design'. The procedure is quite simple and many good programmers use it automatically. Structured programming can be summed up in five words — "think before you write code". There are a number of different techniques of structured programming, but they are all just different ways of helping you get your brain in gear.

In top-down design you start by considering the *whole* problem and working out what you want the design to do. The aim is to break the task down into smaller and smaller steps until you end up with a detailed specification where each step can be converted into a few Basic lines. There's not much point in doing this for small projects (except, perhaps, for practice) since you won't need to take many steps before a solution becomes obvious. For larger projects, though, a well-organised approach is very helpful.

Top-down design brings two advantages: brevity, since it's easy to spot duplicated problems and use the same code for each instance; and simplicity, since you keep thinking about a problem (in smaller and smaller steps) until the code required becomes obvious. Both advantages make the program easier to understand, which means you can test and alter it easily.

If you've ever written a long or complicated program, and then had to produce a succession of different versions, you will probably have noticed that there comes a point when it becomes almost impossible to make further changes. The whole thing becomes so tangled that it's quicker to rewrite it than to alter the old version. The more you think before you program, the less this happens, because you tend to anticipate problems — and work out neat ways around them; you don't end up fiddling with a faulty routine in memory, trying to salvage bits as best you can.

Figure 2. ZIP Compiler structure: first draft

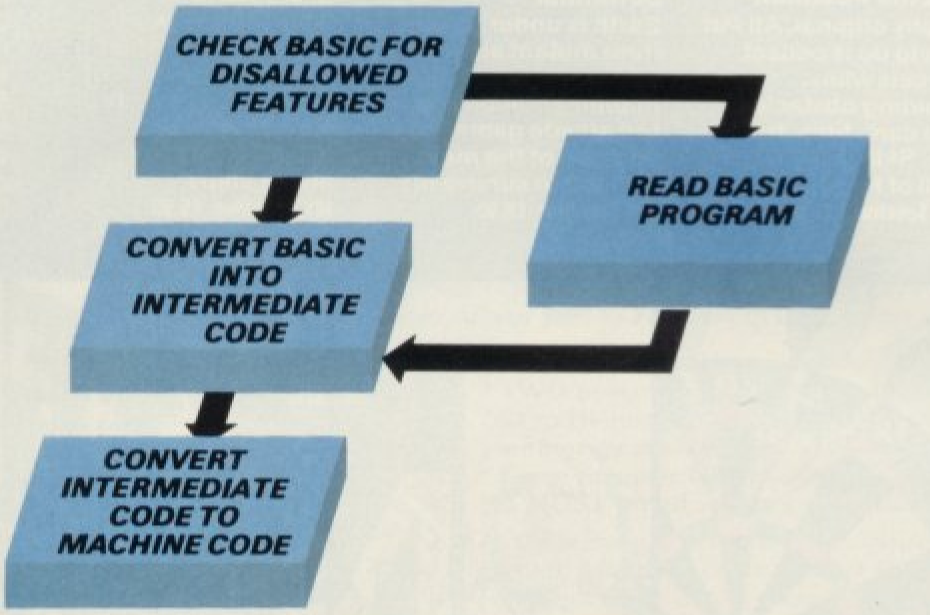
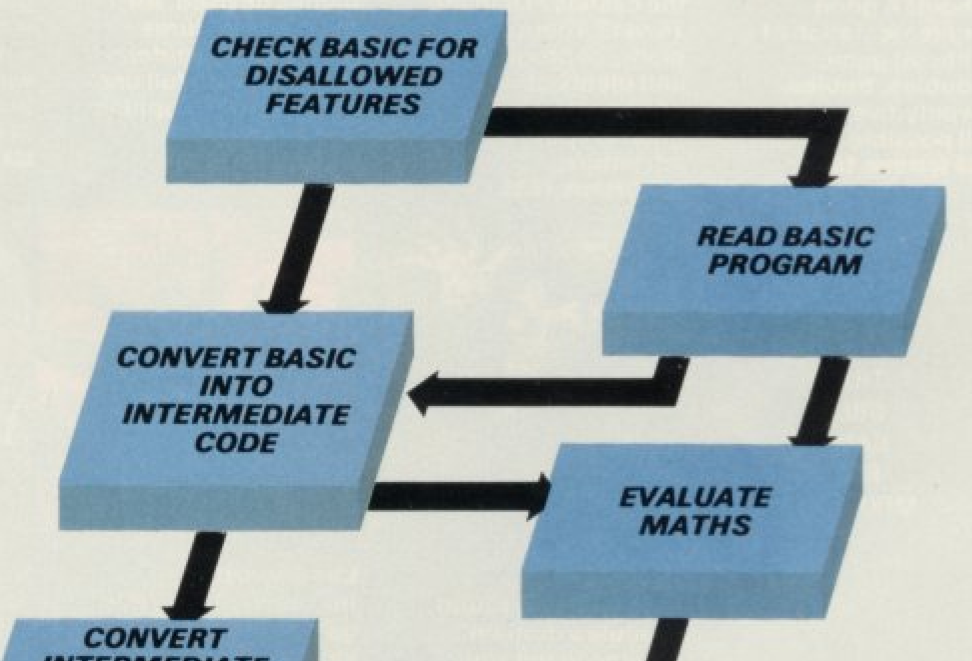


Figure 3. ZIP Compiler structure: second draft



Top-down designs get finished (because you can manoeuvre around problems at the design stage), they do what you intended (because you decided exactly what that was before you started) and they go on working (because you know what the effects of changes will be).

In general, an hour spent designing saves two hours of testing. And in the extreme case, if you spend too long designing you'll get bored and never write the program at all, in which case you don't have to test it. (You knew it would work, anyway!)

DESIGNING ZIP

In the simplest instance, *ZIP* is a program which reads Basic and spits out the equivalent in machine code. We'll refine this definition step-by-step

ADDING ZIP 2!

until we can see the structure of the compiler.

Basic and machine code are two very different things, so it would be a good idea to use some kind of staging-post between the two — an 'intermediate code', as was discussed last month — translate Basic into intermediate code, then convert intermediate code to machine code.

At this point we must consider what we mean by Basic. We don't mean the entire language as listed in Appendix C of the Spectrum manual, because there's lots of it and many statements (such as the complicated arithmetic functions) would be no better performed in machine code. Therefore, we must choose a subset of Basic with two criteria — usefulness (could programs be written without this feature?) and efficiency (would this feature benefit from compilation?).

We end up with the subset listed last month. Don't worry too much about the details because they are not important at this stage. You can add and subtract many features without invalidating the design. The major decisions are to allow whole-number arithmetic, graphics and flow-of-control statements (IF, GO TO, GO SUB, FOR, etc).

Now another program suggests itself — checking that the program to be compiled conforms with our subset. The program can do this as it produces intermediate code, but it's simpler (and therefore probably better) to check for errors in an extra, early, step. In this way we can find errors at the end of a program without having to work out all of the intermediate code up to that point.

We now have a three-stage design: check Basic, convert Basic to intermediate code, convert intermediate code to machine code. Immediately, we spot that there are two steps which read the Basic. They could share some code. Figure 2 shows the resultant structure, with three main steps and subsidiary routine shared between the first two.

Next, we go back to the definition of Basic and think a bit about the second step — converting Basic into machine code. We know the format of each statement: LET statements consist of a variable name followed by an equals sign then an expression (any arithmetic value or calculation); BORDER statements contain just the keyword followed by an expression; and so on. 'Expressions' crop up over and over again — in POKE, PRINT, IF RANDOMIZE and other statements.

Expressions can also be pretty complicated:

```
BORDER ((A>=23)+N(I))/4 AND PEEK
(23767)+USR(32000)
```

The above would be perfectly legal, although rather unlikely — so it seems a good idea to package that up into a separate routine which the second step can call upon whenever it's needed. The new 'maths' routine will need to have access to the Basic program (so that it can scan through expressions) and it will also spit out intermediate codes for each action represented in the expression. Figure 3 shows the resultant structure.

We can continue in this way, spotting problems and assigning 'steps' to solve them, until each step seems trivial. Then we write the program. Rather than consume the rest of this magazine with further steps, we've listed the main parts of *ZIP* in with the listing (see Figure 1). Some of these are common steps which occur in any program — we need an 'initialisation' step to reserve space for variables and code, and a 'main program' step to tie the other sections together.

The other three new sections are concerned with manipulated values in memory (DEEK and DOKE, used by the machine code generator and the Basic reader) and the handling of GO TOs and GO SUBs. Our intermediate code does not contain line numbers so we must use a separate table of numbers and line positions (memory addresses) to record the destination of each GO TO. (Our intermediate code doesn't use line numbers because we want to be able to jump directly from one line to another, without the need to search for the line required.)

The only difference between this design and the final program is the fact that Lexical Analysis performs most of the checks which we have assigned to PASS 1. This is done because it turned out to be easier to trap errors as soon as they were found than to pass on messages about them.

THE PROGRAM — FINER POINTS

We could probably write a book on the way the *ZIP* program works — it took four months to write and our notes are about 300 pages long — but the resulting volume would probably be something of a minority interest! The program has been written to be read, with comments in key places and manifest constants whenever possible. The rest of this article explains some of the more devious parts of the listing.

It's quite simple to use *ZIP* without knowing how the compiler works. Load the *ZIP* program, then type in (or MERGE) the program you wish to compile, making sure that it occupies less than about 13K and uses no line numbers greater than 4999. Type GO TO 5000 to start *ZIP*. The library is loaded from tape and the compiler starts.

If you had already loaded the library you could start *ZIP* with the command GO TO 5035, or GO TO 5800 if you haven't typed CLEAR since you last used *ZIP*. The program will scroll up the screen as *ZIP* checks it for obvious errors. A message appears for each error, immediately after the error has been found. If the entire program is scanned without errors, *ZIP* begins to generate machine code.

The current line number and number of bytes of machine code produced is shown as compilation takes place. During this second phase more intricate errors may be detected. Once again the sarily performed in the order they are written. For example, think of the statement:

```
PRINT 2+3*4
```

We can't evaluate the expression 2+3*4 from left to right — otherwise we'd end up with the answer 20 (5*4). If you try this command on the Spectrum you'll get the correct answer 14, because the computer performs multiplications before addition. Chapter 3 of the Spectrum introductory booklet explains this.

Basic uses 'algebraic notation', which means that it performs calculations in brackets first, then multiplications and divisions, then addition and subtraction, just as you were taught in school algebra lessons (boo!). The stage at which an operator (a sign such as '+' or '*') is performed is called its 'priority'. Multiplication has a higher priority than addition, so '*' is performed before '+'.

This problem becomes even more complicated when you remember that expressions can contain brackets, array references (with expressions inside them), functions like PEEK and ATTR, plus variables and numbers. You can even use comparison operators such as '>=', '<>', 'OR' or 'NOT', and so forth, in Spectrum arithmetic — try 'PRINT 1>2' or 'PRINT 3 AND 2' if you don't believe this.

ZIP CALCULATIONS

It takes quite a complicated program to resolve an expression and produce all of the steps — in the right order — to work out the correct value. The section of *ZIP* between lines 6700 and 6925 is used to evaluate expressions. It was written by the ingenious Jon Smith, who also wrote much of the PASS 2 routine.

The expression routine, called 'maths', will handle 27 different operators. The operator '—' appears twice, since it can be used two different ways, as in the expression 'PRINT 2--2'. The first minus is a subtraction sign, the second so-called 'unary' minus marks the second '2' as negative number. Of course, PRINT 2----2 would be valid too. If you use a whole line of minus signs you clog up the program and receive a 'CALCULATION TOO COMPLEX' error, but this should not be problem in normal use!

ZIP re-orders arithmetic expressions into a form of 'Reverse Polish notation' (RPN) — the calculations

ADDING ZIP 2!

sequence used by Hewlett-Packard calculators and Forth programmers. RPN differs from Basic's 'Algebraic notation' in that the values to be manipulated always precede the operators. In the RPN you write '2 2 +' rather than the familiar '2 + 2'. The RPN expression means take two, take two again, then add what you've got. This leaves four — the correct answer.

The plus operator always adds the last two values together, leaving one value. Multiply, divide and subtract work the same way. PEEK replaces the last value with the contents of the address it specifies. The point is that *any* expression can be written in RPN and evaluated correctly from left to right, without the need for brackets. '2 + 3 * 4' becomes '2 3 4 * +'. The only problem is converting them between the two formats.

ZIP scans through an expression, generating instructions to put each can't say 'GO TO variable' in a compiled program — *ZIP* needs to know the destination as the code is generated — but the speed-up is usually well worthwhile.

In order to compile GO TOs and GO SUBs, *ZIP* must keep a list of line numbers and addresses of the code for each line. This list is kept in a table between TOP and LINETAB, above the compiled program. The variable XREF (cross-reference) indicates the bottom of the table, which slowly fills up as each line is compiled and its address is entered (by subroutine ADDLN).

Whenever *ZIP* finds a GO TO to a previous line, it searches through the table for the line number concerned and extracts the required address. The search is performed by the FIND LINE subroutine, which uses a simple but effective technique called a 'binary chop'. Rather than scan through the list from the start, FIND LINE starts in the middle and compares that entry with the line number required. If the entry is too high, FIND LINE looks at the entry a quarter of the way down — otherwise it looks three-quarters of the way through the list. If the entry there is too low it skips forwards to the midpoint of the last two checks; otherwise it skips back a similar amount.

The effect is that the list is cut in half, then half again, and so on, until the line (or the nearest subsequent value) is found. This technique can find an entry in a list of 500 lines in just nine steps. FIND LINE copes well with jumps to previous lines, but it can't handle 'forward jumps' — GO TOs and GO SUBs which have not yet been compiled.

Another problem involves IF statements. *ZIP* compiles these using the rule 'IF (expression) is false THEN GO TO (this line +1) ELSE (carry on with this line)'. The effect is that the rest of the line is skipped if the expression fails, which simulates the Basic IF statement, but this also involves a 'forward jump'. How can we generate the 'GO TO (this line+1)' when we've not finished scanning the current line yet, so we've no idea where the next line will start?

ZIP solves these problems by deferring them. Since we don't know where the destination line is yet, we'll have to put off generating that bit of code until we do. Instead of a line address *ZIP* puts the number of the line required into the machine code, for the time being. *ZIP* also makes an entry into a 'line request table' between LINETAB and TP, and then it carries on as if nothing went wrong. The entry is the address at which the dummy line number was stored — the address which must be corrected later.

Once the entire program has been scanned, the address of every line is known, so FIND LINE always works. A simple routine called PATCH now scans through the line request table. It reads the line number required from each address to be patched up, and uses FIND LINE to convert the number into a real address. Hey presto! All of the forward jumps are resolved.

GUILTY EXPRESSIONS

As we said earlier, mathematical expressions appear all over the place in Spectrum programs. These present special problems when writing a compiler, because the steps involved aren't necesline number and a full error message appears, but *ZIP* skips to the next line of Basic as soon as one of these errors is found in a line.

If the compilation takes place without any errors the message 'ZIP HAS FINISHED' appears, followed by the command needed to save the entire program — library, variables and code — and the command to RUN the machine code. Normally the command 'RANDOMIZE USR prog' will run the code. (The value of 'prog' is shown on the same line, in case you type CLEAR or overwrite the value of 'prog'.)

Most of the error messages are self-explanatory; if you are confused, check the specification published last month. A full list of the messages, with explanations, is also contained in the *ZIP* compiler cassette package.

Figure 4 lists the main variables used in *ZIP*. The program uses a large number of 'manifest constants', which are not listed — these are values which would normally be typed as numbers, but we have written them as variables to make the program more readable. For instance, we use the constant 'CAPITAL Z' instead of 90 (the ASCII code of 'Z'), and type 'GO TO EXPNEXTS' when we mean 'get the NEXT S-ymbol in the EXP-ression'.

HIDDEN NUMBERS

ZIP takes advantage of the way Basic stores numbers in a program. As well as the individual digits of each number, Basic stores an 'invisible' binary copy of each number in a program line. The copy occupies six bytes. In line 5060 we've used variables 'Y' and 'N' instead of the constants '1' and '0', and saved about 500 bytes in the process. This duplication is designed to speed up ZX Basic by removing the need to convert numbers from digits into binary, but in practice it wastes memory to little effect.

The first invisible byte is a CHR$ 14 which tells LIST to skip the next five bytes. Then comes the number in binary. Decimal numbers, and very large whole numbers, use all five bytes, but the small integers allowed by *ZIP* are kept in a compressed form. The *ZIP* subroutine at line 6110 is used to extract such numbers from a program.

The compressed form is marked by a zero byte immediately after the CHR$ 14 byte. If this byte contains any value other than zero, the number is not allowed by *ZIP*, and an error message is produced. The fourth and fifth bytes contain the number, which can be read using the double-length PEEK subroutine, DEEK. Finally the third byte is checked — if it's not zero, the number is negative.

Even binary numbers such as BIN 10101010 are stored in this format, which is why *ZIP* lists binary numbers in decimal as it checks the program. The only numbers *not* stored in this way are program line numbers. These are prefixed with CHR$ 13 (Enter) and writ-

SCANNING THE EXPRESSION '2*3+4/-4'

VALUE STACK	OPERATOR STACK	EXPRESSION LEFT
empty	END	2*3+4/-4
2	END	*3+4/-4
2	*END	3+4/-4
32	*END	+4/-4
The '*' is performed as it has higher priority than '+'.		
6	+END	4/-4
46	+END	/-4
46	/+END	-4
46	-/+END	4
446	-/+END	
The '-' is performed as it has higher priority than 'END'.		
-446	/+END	
The '/' is performed as it has higher priority than 'END'.		
-16	+END	
The '+' is performed, and both ENDs meet — we've finished!		
5		

ADDING ZIP 2!

ten in reverse order, high byte first. Line 6100 is used to read line numbers.

JUMPING AHEAD

ZIP performs GO TO and GO SUB statements at least a thousand times faster than ZX Basic. *ZIP* uses direct machine code jumps, without searching through a table of lines. This means you number or variable value on a 'stack'. This is a kind of one-ended list — you can put values into the list or take them out, but you can only ever take out the most recently-entered value. The effect is rather like pushing oranges into an elastic sock — to get at the first orange pushed in, you must remove all of the subsequent ones in the opposite order to that in which they were inserted.

If you put the numbers '1','2','3' and '4' into a stack and then retrieve them one by one, you get '4' back first (the most recently 'pushed' value), then '3', '2' and '1' in descending order.

As *ZIP* stacks up the values in an expression it also makes a stack of the operators, but with one crucial difference — if the previous operator (the one on the stack) had as much or more priority as the operator which has just been found, *ZIP* immediately generates the code to perform that previous operator and takes it off the stack.

This process continues until the operator at the top of the stack has a lower priority than the one which has just been found. Then the new operator is put onto the stack and the scan continues. A special 'end marker' with a very low priority is put on the stack at the start, so that *ZIP* knows when it has run out of

MAIN VARIABLES USED BY ZIP

A()	Size of each compiled array during PASS 1 — address of array storage thereafter: A(1) for compiled array A, A(2) for array B, etc.
ARRAYS	Address of the start of compiled array storage.
ASSMOD	Assignment mode — the number of the machine code template needed for an assignment, or temporary storage for a template number.
ASSVAR	Assignment variable: the number (0=A, 25=Z) of the variable of array to be loaded.
BOTTOM	Address of the start of the library (see the relevant table last month for the ZIP memory map).
BUGS	The authors' pet rabbit. Also indicates the type of error found during PASS 2.
C	Number of the machine code template requested (to save space, constants were not used to template numbers — see the relevant table last ussue for a list of templates in order, eg. C=4 for GO TO).
EOF	The first line number not to be compiled.
ERRORS	The number of errors found.
L()	L (Z) is the most recent operator found in an expression which has not been performed yet.
LAST	The previous token.
LINES	Number of Basic lines found.
LINETAB	Address of the end of the line address table.
LNUM	Line number of the most recent Basic line scanned.
NUM	A number returned with S: if S=QUOTE, NUM is the address of the first character in a string; if S=ENTER, NUM contains the new line number; if a constant value has been found, S=NUMBER and NUM contains the value concerned.
O()	The intermediate code associated with a given arithmetic operator or function.
P	The next location in the program to be scanned.
P()	Priority of each operator — the higher the value of P(op) the later the operation is performed in a calculation.
PC	Address of the last machine code generated.
PEEP	Flag used for 'peephole' optimisation (see next month's article).
PROG	Address of the start of the machine code.
S	The current symbol, usually an ASCII code.
S()	S (Z) is the variable number (0—25) of the most recent array indexed in an expression (only used when L (Z) = INDEX).
SCROLL	System scroll counter (see Spectrum manual, P175).
SOURCE	Address of a pair of locations holding the address of the start of the Basic program.
T	The variable number (0—25) or constant passed for code generation; also various miscellaneous uses.
T()	Flag indicating whether or not character N is allowed by ZIP, where T (N-KEYWORD) is zero if the keyword is not allowed.
TOP	Address of the last location used by ZIP.
TP	Address of the end of the line request table.
VARS	Address of the start of variable storage.
XREF	Address of latest entry in line address table.
Z	Subscript used to indicate the 'most recent item' on the expression evaluation stack.

operators. Another 'end marker' is produced at the end of the expression. When the two markers meet we know that the whole expression has been evaluated.

If this seems like gibberish, look at Figure 5, which should be worth at least 1e3 words! It shows the process of scanning the expression '2*3+4/−4'. The result, '5', is left on the value stack.

Brackets are handled like the END markers, but with a slightly higher priority. The effect is that expressions in brackets are worked out before the operators around them are applied. The open bracket works as a kind of 'block' on the stack, which only a close bracket can knock off. Even the brackets of an array subscript can be handled in this way. *ZIP* uses a special operator 'index' to mark the state of a subscript calculation.

Examine the expression evaluation routine, using the variable information in Figure 3. The subroutine EXPUSH is used to stack an operator, and EXPOP removes one entry (by moving Z down so that the stack is one unit shorter). One extra quirk is line 7802, which checks for the special case of a line minus sign before a number. If possible it generates a negative number immediately, rather than the two steps 'stack number' and a 'make last number negative'.

'OH FOR A QL' DEPT

The *ZIP* compiler is not a perfect example of structured programming; it's been

ADDING ZIP 2!

deliberately crammed into 16K of memory (plus the library) and written in Basic. Squashed as it is, *ZIP* does work, and it shows that you don't need a disk system or a 'fashionable' language to write a complex, useful software tool. The basic principles of structured design have been followed — it's very doubtful that the program would ever have been finished otherwise!

BUGLETS

There's no way to be sure that a compiler is totally 'bug free' — there are just too many possibilities! We've tested *ZIP* on dozens of programs but we may still have missed some flaws. If you find a bug please write to *Your Spectrum* stating the exact circumstances in which it occurs (with a listing if possible) and we'll do our best to fix it quickly for you.

NEXT MONTH We'll conclude this series with a look at possible enhancements to the compiler. We'll provide subroutines to simulate INKEY$ and RND, plus 'worked examples' showing you how to add high resolution PLOT and DRAW commands. There'll be a detailed example of the way the machine code templates are fitted together, and an example program to compile which shows just how good ZIP can be!

The ZIP Compiler

This selects combinations of those blocks which work the same way as the Basic program — only much faster!

Lines 5000-5790 Initialisation. This reserves space for arrays and sets up constants for use later in the program.

```
5000 REM Spectrum BASIC compiler
5005 REM Last revision: 22/03/84
5010 REM S N Goodwin & J A Smith
5015 REM
5020 REM #### INITIALISATION
5025 CLEAR 53246
5030 PRINT "Load library.": LOAD ""CODE
53247
5035 DIM t(91): REM Token types
5040 DIM a(27): REM Array bases
5045 DIM p(27): DIM o(27): REM Function
precedences & opcodes
5050 LET maxz=22: DIM l(maxz): DIM s(max
z): REM Maths stacks
5060 RESTORE : LET y=1: LET n=0: FOR i=1
 TO 91: READ t(i): NEXT i: DATA n,n,n,n,
n,n,y,y,y,n,n,n,n,n,n,n,n,n,n,n,n,y,n,y,
y,y,y,y,n,y,y,y,y,y,y,y,y,n,y,y,y,n,n,n,
n,n,n,n,n,n,n,n,y,y,y,y,y,y,y,n,n,y,n,n,
n,n,y,n,y,y,y,y,y,y,n,n,y,y,y,y,y,n,n,n,
y,y,y,n,y,y,n
5070 FOR i=1 TO 27: READ p(i): READ o(i)
: NEXT i: DATA 14,-1,0,-1,8,15,6,16,1,-1
,6,17,9,56,8,18,12,3,5,42,5,51,5,44,11,3
9,11,40,11,36,11,38,11,37,-1,-1,11,47,4,
41,-1,-1,2,54,3,55,5,43,5,45,5,46,11,50
5190 REM
5200 REM #### MANIFEST CONSTANTS
5205 LET true=1: LET false=0: LET eof=50
00
5210 LET appos=39: LET plus=43: LET numb
er=14: LET enter=13: LET string=50: LET
overflow=15: LET quote=34: LET comma=44:
 LET space=32: LET dollar=36: LET hash=3
5: LET op=42: LET variable=1
5220 LET keyword=164: LET digit 0=48: LE
T digit 9=57: LET capital A=65: LET capi
tal Z=90: LET colon=58: LET semi=59
5225 LET attr=171: LET at=172: LET tab=1
73: LET chr=194: LET then=203: LET to=20
4: LET step=205: LET ink=217: LET paper=
218: LET flash=219: LET bright=220: LET
inverse=221: LET over=222
5230 LET border=231: LET dim=233: LET re
m=234: LET for=235: LET goto=236: LET go
sub=237: LET input=238: LET let=241: LET
 pause=242: LET next=243: LET poke=244:
LET print=245: LET random=249: LET if=25
0: LET cls=251: LET clear=253: LET retur
n=254
5235 LET out=223: LET stop=226: LET int=
186: LET bin=196: LET lbracket=40: LET r
bracket=41: LET small a=97: LET small z=
122
5250 LET source=23635: LET scroll=23692:
 REM System variables
5255 LET index=9: LET uminus=7: LET opbr
t=1: LET clbrt=2
5390 REM
5400 REM #### BASIC LABELS
5405 LET gets=7986: LET fetch=6015: LET
init=6010: LET dump=8035
5410 LET deek=6205: LET doke=6300: LET f
ind line=6745: LET too big=6715: LET req
uest=6900: LET skip st=7970: LET p2error
=7950
5415 LET pass1=6500: LET pass2=7000: LET
 patch=6795: LET tables=6700: LET addln=
6725: LET next1=6555: LET Z80=8000: LET
itm=7330
5420 LET p1error=6125: LET parsestr=7910
: LET nextln=7030: LET nextst=7050: LET
pre colon=7660: LET atcolon=7670: LET st
ore=7690
5440 LET exppush=7752: LET exppop=7770:
LET maths=7785: LET expnexts=7900: LET e
xpnext=7800: LET expop=7822: LET expdoop
=7857
5790 REM
```

Lines 5800-5990 Main program. This performs a compilation by calling other routines.

```
5800 REM #### MAIN PROGRAM
5805 LET bottom=53247: LET vars=bottom+1
537: LET arrays=vars+2848: LET prog=arra
ys: LET pc=prog
5810 LET top=65535: LET linetab=top: LET
 tp=linetab
5815 LET last=0: LET errors=0: LET lines
=0: FOR i=1 TO 26: LET a(i)=0: NEXT i
5820 GO SUB pass1: PRINT  PAPER 5; INK 0
;'lines;" LINES scanned: ";errors;" ERRO
RS.": IF errors THEN  STOP
5830 GO SUB tables: GO TO pass2
5840 IF errors THEN  PRINT AT 0,0; PAPER
 1;"ZIP FOUND ";errors;" ERRORS.": STOP
5845 GO SUB patch: REM Correct forward j
umps
5850 PRINT AT 0,0; PAPER 1;"ZIP HAS FINI
SHED    (";pc-prog-36
5860 PRINT AT 19,0; INVERSE 1;"Save: SAV
E ""???""CODE ";bottom;",";pc-bottom+2,,
,"Run:   RANDOMIZE USR prog [";prog;"]":
 STOP
5990 REM
```

Lines 6000-6190 Lexicon Analysis. This reads the Basic program, returning symbols one by one as they are encountered.

```
6000 REM #### LEXICAL ANALYSIS
6005 REM Init & Fetch (source)
6010 LET t=source: GO SUB deek: LET p=t:
 LET s=enter: GO TO 6100
6015 LET last=s: LET s=PEEK p: LET p=p+1
6020 IF s=enter THEN  GO TO 6100
6021 IF s<=space THEN  GO TO 6015
6022 IF s>=small a THEN  IF s<=small z T
HEN  LET s=s-32: REM lower case
6023 IF s=quote THEN  LET num=p: RETURN
6024 REM Analyse variable name
6025 IF last<capital A OR last>capital Z
 OR s<digit 0 OR s>capital Z THEN  GO TO
 6030
6026 IF s>digit 9 THEN  IF s<capital A T
HEN  GO TO 6030
6027 PRINT CHR$ s;: LET s=PEEK p: IF s>=
small a AND s<=small z THEN  LET s=s-32
6028 IF (s>=digit 0 AND s<=digit 9) OR s
=space OR (s>=capital A AND s<=capital Z
) THEN  LET p=p+1: GO TO 6027
6029 LET s=variable: GO TO p1error
6030 IF s>=digit 0 THEN  IF s=bin OR s<=
digit 9 THEN  GO TO 6110
6040 IF s=int THEN  GO TO 6015
6045 IF s=hash OR s=dollar THEN  GO TO p
1error
6050 IF s>keyword THEN  IF t(s-keyword)=
0 THEN  GO TO p1error
6090 RETURN
6095 REM New line found
6100 LET num=256#PEEK p+PEEK (p+1): LET
lnum=num: LET t=p+2: GO SUB deek: LET p=
p+4: RETURN
6105 REM Number found
6110 IF PEEK p<>number THEN  LET p=p+1:
GO TO 6110
6112 IF PEEK (p+1)<>0 THEN  LET s=overfl
ow: LET p=p+6: GO TO p1error
6115 LET p=p+3: LET t=p: GO SUB deek: LE
T num=t: IF PEEK (p-1) THEN  LET num=-nu
m
6120 LET p=p+3: LET s=number: RETURN
6122 REM PASS 1 Error (p1error)
6125 PAPER 6: INK 0: PRINT
6130 IF s=dollar THEN  PRINT "Strings";
6135 IF s=variable THEN  PRINT "Variable
 name";
6140 IF s=hash THEN  PRINT "Streams";
6145 IF s>keyword THEN  PRINT CHR$ s;
6150 IF s=overflow THEN  PRINT "Decimal
values & integers beyond+/-65535";
6155 PRINT " not allowed.": LET errors=e
rrors+1: PAPER 0: INK 7: RETURN
6190 REM
6195 REM #### DEEK and DOKE
```

Lines 6200-6490 DEEK and DOKE. These are general purpose routines used to store and retrieve 16-bit values (0-65535) in memory.

```
6200 REM Deek - t=contents of address t
6205 LET t=PEEK t+256#PEEK (t+1): RETURN

6295 REM Doke - put t at address i
6300 POKE i,t-INT (t/256)#256: POKE i+1,
INT (t/256): RETURN
6490 REM
6495 REM #### PASS1 Allocate RAM
```

Lines 6500-6690 PASS 1. This routine checks that the symbols returned by Lexicon Analysis are allowed by ZIP.

```
6500 PAPER 0: INK 7: BORDER 0: CLS : PRI
NT  PAPER 5; INK 0;"SPECTRUM BASIC COMPI
LER  PASS 1.": GO SUB init
6505 IF s=enter THEN  IF num>=eof THEN
RETURN
6507 IF s=enter THEN  LET lines=lines+1:
 PRINT : PRINT num;" ";: POKE scroll,255
: GO TO next1
6510 IF s=number THEN  PRINT num;: GO TO
 next1
6512 IF s=rem THEN  GO TO 6545
6513 IF s=dim THEN  GO SUB 6565
6515 IF s<>quote THEN  PRINT CHR$ s;: GO
 TO next1
6520 PRINT """";
6530 PRINT CHR$ PEEK p;
6535 IF PEEK p<>quote THEN  LET p=p+1: G
O TO 6530
6560 REM Evaluate DIM
6565 PRINT " DIM ";: GO SUB fetch: IF s<
capital A OR s>capital Z THEN  GO TO 659
0
6570 LET v=s-64: PRINT CHR$ (s);: GO SUB
 fetch: IF s<>lbracket OR a(v) THEN  GO
6540 LET p=p+1: GO TO next1
6545 PRINT CHR$ s;: LET s=PEEK p: LET p=
p+1: IF s<>enter THEN  GO TO 6545
6550 LET p=p-1: GO TO next1
6555 GO SUB fetch: GO TO 6505
TO 6590
6575 PRINT "(";: GO SUB fetch: IF s<>num
ber THEN  GO TO 6590
6580 PRINT num;: GO SUB fetch: IF s<>rbr
acket OR num<1 THEN  GO TO 6590
6585 LET a(v)=num: RETURN
6590 PRINT  PAPER 6; INK 0;'"Faulty DIM
statement.": LET errors=errors+1: RETURN

6690 REM
6695 REM #### BUILD RAM TABLES
```

Lines 6700-6718 Build RAM Tables. This allocates space for line cross-references and arrays.

```
6700 FOR i=1 TO 26: LET t=prog: LET prog
=prog+a(i)#2: LET a(i)=t: NEXT i: LET a(
27)=prog: LET pc=prog
6705 LET linetab=top-4#lines-4: LET tp=l
inetab: FOR i=tp TO top-1: POKE i,127: N
EXT i
6710 IF tp>pc THEN  RETURN
6715 PRINT AT 21,0; PAPER 6; INK 0;"Insu
fficient memory.": STOP
6718 REM
```

Lines 6720-6990 Cross-Reference Routines. These routines manipulate the cross-reference tables used to resolve IFs, GO TOs and GO SUBs.

```
6720 REM #### CROSS REFERENCE ROUTINES
6725 REM Add an entry to line address ta
ble (addln)
6730 LET t=lnum: LET i=xref: GO SUB doke
: LET i=xref-2: LET t=pc+1: GO SUB doke:
 LET xref=xref-4: RETURN
6740 REM Find line n (return T pointing
to its address)
6745 LET L=1: LET u=(top-linetab)/4
6750 IF L>u THEN  LET t=u#4+linetab-2: R
ETURN
6755 LET t1=INT ((u+1)/2): LET t=t1#4+li
```

69▶

Your Spectrum ***is continuing to scour the country's computer clubs to get the undiluted thoughts, feelings and impressions of dedicated games enthusiasts — in order to provide our readers with unbiased evaluations. Any club wishing to offer its reviewing services should contact*** **Ron Smith, Spectrum Soft, 14 Rathbone Place, London W1P 1DE.**

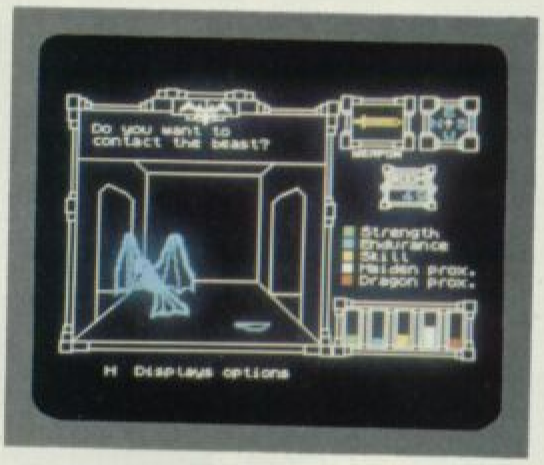
We sent this month's batch to the Bangor Computer Club of North Wales, where their expert reviewers got to work; their plaudits and barbs can be seen over the following pages. First, let's have a look at the club and what it offers its members.

The Bangor Computer Club was formed 18 months ago by noted author, Dilwyn Jones. He wanted to meet the needs of local home computer users, and holds meetings in member's homes on a spur-of-the-moment basis. There are currently 25 to 30 members whose ages range from 13 to 30 and, although there's no longer a fixed membership fee, those attending are expected to splash out towards the cost of the evening.

Its members have the chance to try out each other's software and exchange ideas. Members can always call upon their colleagues to help solve problems. Anyone in the area who's thinking of joining the club should contact Dilwyn on 0248 354023, outside office hours.

THE LINE-UP

This month's reviewers are Ieuan Davis (25), Gerralt Jones (22) and Brian Pedlar (31); all three are long-standing members of the club and avid games enthusiasts to boot. They represent the 'Old Gold' strata of Speccy users, so we won't be expecting starry-eyed reactions to our selection of new games.



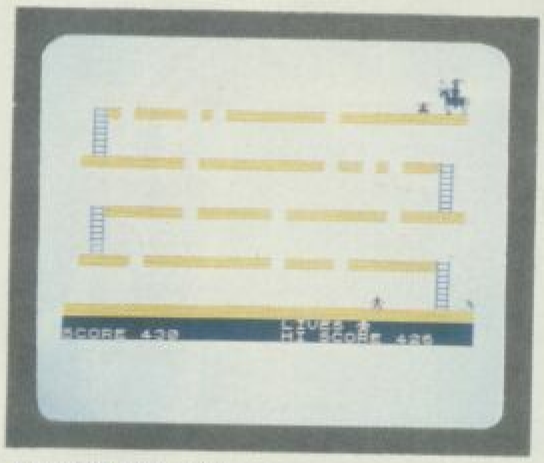
DRAGONSBANE
Quicksilva/£6.95

A graphic adventure through 172 rooms; avoid 40 monsters while trying to rescue the helpless and hapless Princess Paula. The usual assortment of monsters, both friendly and otherwise — and you can also use joysticks.

Gerralt Slow, with lots of 'Y/N' single key answers. Good use of colours, but it doesn't look too good if all you've got is a monochrome TV. *6/10*

Ieuan The game can be crashed quite easily — which is a bit disappointing. It also takes a lot of playing before you get to understand the characteristics of the monsters. *5/10*

Dilwyn Good clear graphics. It uses all the 48K of available memory, and is jolly good fun. *8/10*



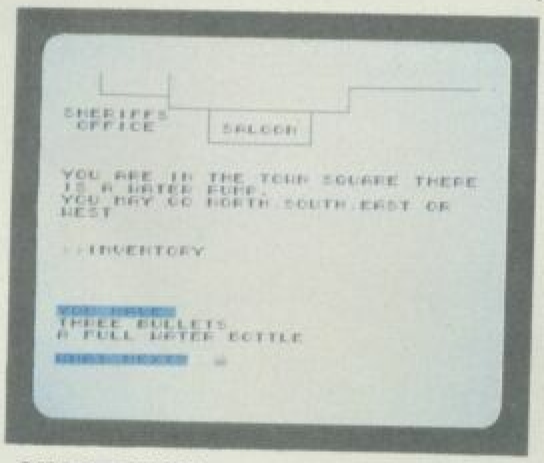
KILLER KNIGHT
Phipps Associates/£5.00

A killer knight has captured your companion, so naturally it's your duty to carry out a rescue. Unfortunately, KK is not above doing everything in his power to stop you reaching the tower. . .

Dilwyn For some reason I expected this to be a text-only adventure, when actually it's a machine code graphic arcade game. I didn't go for the choice of keys — but then it's also possible to use a Kempston joystick. *7/10*

Ieuan A 12th century blend of *Donkey Kong* and *Digger*. It's quite fast, indeed sometimes too fast. The knight moves and sounds like a cricket, and I also found that jumping slows down the graphics. *7/10*

Gerralt Only a highest score is shown on-screen — you're not even given the satisfaction of seeing your name in lights! *7/10*



GHOST TOWN
Virgin Games/£5.95

A Wild West text-based adventure game set in a deserted American town after the California Gold Rush. You get line maps of streets and buildings, and the whole thing is written in Basic — there's no machine code in them thar hills, pardner!

Ieuan *Ghost Town* has an acceptable response time to entries, but there's no variation — so once you've solved it, well, that's it. There's only minimal use of sound. *5/10*

Gerralt The game is set in a fairly small town, so it doesn't take you long to find out what's going on there. The blue and cyan colours are nice — even contrasting well on a monochrome TV. *6/10*

Brian A fairly standard adventure game, but with a reasonably large vocabulary. *5/10*



BALLOONING
Heinemann/£9.95

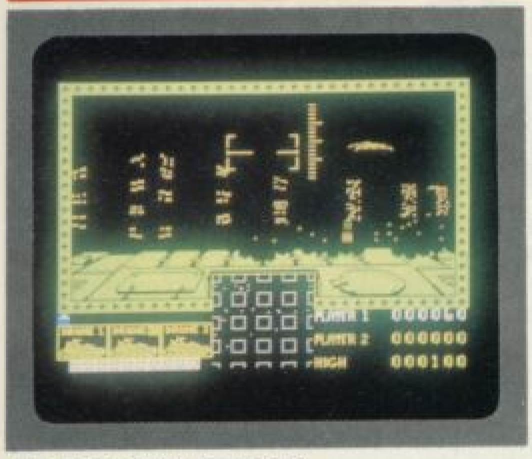
An educational game for Spectrum teenies aged eight-to-12 which sets out to teach observation, experimentation and the recording of data. A keyboard overlay for most controls is provided and there's an accompanying activities booklet.

Dilwyn Quite good graphics of the balloon and the instrument panel. A nice program, but at nearly £10, slightly expensive for what it is. *7/10*

Ieuan This package is pitched just about right — I think it's worth the money. Speed is adequate, but then it's not meant to be Brands Hatch! It makes education fun. *8/10*

Gerralt A great pity there's no accompanying sound with the game. The activities booklet is a good idea, but I did find the numbered menu a bit confusing. *7/10*

SPECTRUM SOFT



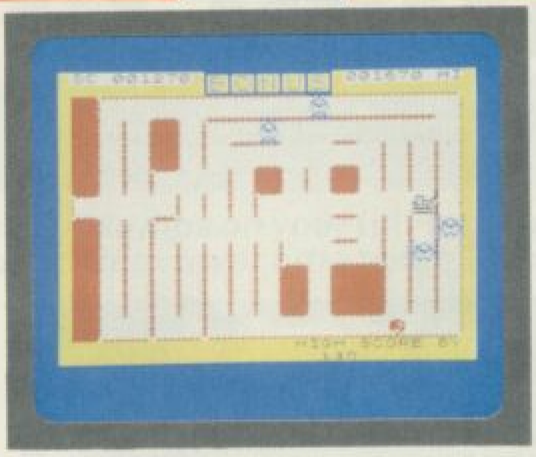
3D SEIDDAB ATTACK
Hewson Consultants/£5.95

Patrol the city streets and later the countryside destroying any Seiddabs that come into range. You control both surface-to-air missiles and a short-range radar display.

Gerralt The 3D effect is quite realistic and adds a lot to the game. The speed is good too — not too fast and not too slow. I liked the split screen display, although the idea is not all that new. *8/10*

Brian You can use a Kempston joystick with this game, which certainly makes it easier to play. It's very addictive, and some useful hints for strategy are given in the sleeve instructions. *9/10*

Ieuan It really needs a colour display as the monochrome fails to show up the necessary details. After a while, my eyes felt a bit strained trying to make out all that was happening on-screen. *7/10*



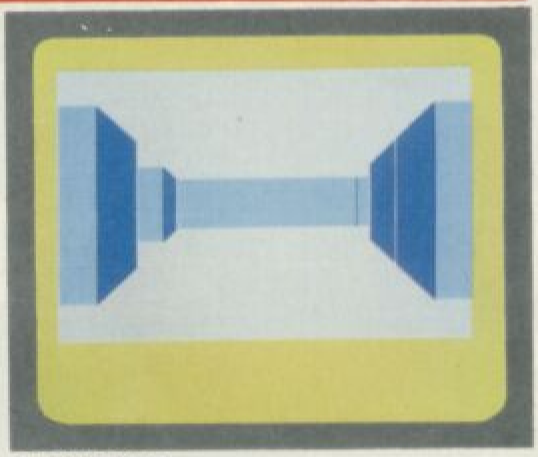
HARD CHEESE
DK'Troniks/£4.95

This one is an arcade game where you have to eat your way through a maze avoiding the deadly bulldogs and zapping them with the deadly virus. Extra points can be gained if you're at all adept at handling cherries!

Brian This is not a new concept, but it's very playable and quite fast — if anything a bit too fast. Unfortunately, there's nothing other than loading instructions on the sleeve. *7/10*

Ieaun It's a good game, with enjoyable sound that doesn't slow things down at all. A joystick is handy for games of this speed; a very, very addictive game. *9/10*

Dilwyn Player keys in this game can be re-defined, which seems to be the ultimate answer to joystick compatibility. When you enter in the high score name, letter descenders disappear! *8/10*



LABYRINTH
Axis/price TBA

Labyrinth is a graphical 3D maze program for all who enjoy getting lost in a good game (ouch!). The plot is the simplest possible — just choose the maze size from four-by-four to 10-by-15, and off you go. You can give up at any time or you can ask for a brief glimpse of the mess you're in.

Dilwyn A great game, which could probably have been expanded to form an adventure. The graphics are still impressive even though they've been around since '82. *9/10*

Gerralt 200 moves is the most allowed. Mostly Basic, the important fast bits are in machine code. However, drawing a 'help' plan is very slow. *8/10*

Brian It could have made a reasonable adventure it there had been some treasure or monsters to watch out for. The graphics are pretty good. *7/10*



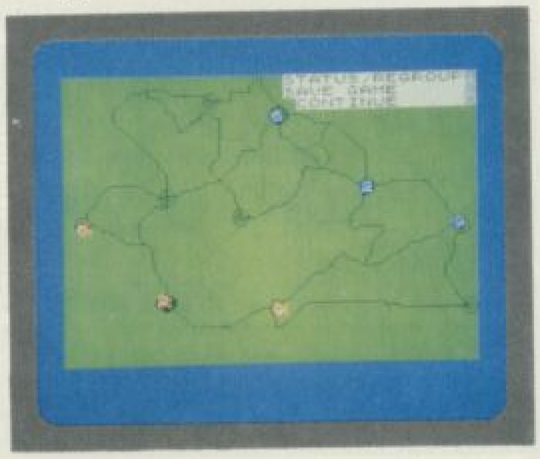
THE ADVENTURES OF ST BERNARD
Carnell Software/£5.95

Every day, Brandy the St Bernard dog has to face blizzards and other hinderances on his way to rescue travellers. His current adventure is to rescue his mistress from the Abominable Snowman.

Gerralt It's rather a slow game, not only is there sluggish response to the controls, but the animation is also jerky. The graphics are reasonably good on the characters and the landscape, but on the whole I found it rather boring. *6/10*.

Ieuan Good graphics detail for the characters and the arctic landscape, but the animation is rather jerky. I found it more pleasing to the eye in monochrome. *7/10*

Dilwyn Players tend to become frustrated at having to go through the first rather silly stage, every time the game re-starts. But overall, not too bad. *8/10*



WAR 70
CCS/£5.50

A Napoleonic wargame set in the 18th Century for two players. The object is to capture your opponent's capital city by occupying it with your army for three days. This game took second prize at the Cambridge awards back in '83.

Gerralt Fairly good graphics, although perhaps they could have been a bit clearer; also the tokens are a bit small. The long explanatory sheet should have carried more details. *7/10*

Dilwyn It sometimes takes a long time to set up the positions — but the use of colour for the various positions is quite good. *7/10*

Ieuan It's a game that could take all day; it's certainly very involved — good for the war games enthusiast! At least the close of play position can be saved to tape and the game continued another day. *8/10*

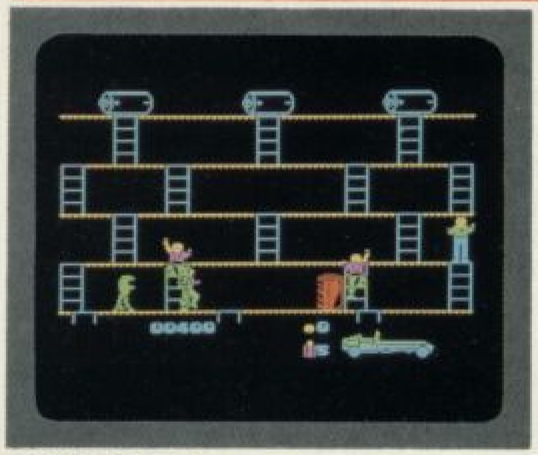


MICRO DRIVIN'
Softel/£6.95

This is a combined text and graphic adventure game where you have to drive a car to compete in the great Island Treasure Hunt. The graphics appear to be very good, apart from the flashing border which tends to rather distract your attention from the instructions which appear on the screen.

Unfortunately, we all found that using certain entries caused the computer the crash and lock out the keyboard! That's more or less all we can say about the game, because after four hours of playing *Micro Drivin'* we couldn't even manage to get the car to move off the starting line. We thought we'd better have a good read of the listing, but even that failed to solve the problem!

By the way, don't be fooled by the title, this program is *not* Microdrive compatible.



ESKIMO EDDIE
Ocean Software/£5.90

Arctic thrills and chills with Percy, Growler and the Snowbugs. First of all you have to rescue Percy the penquin and then battle against the Snowbugs, using the available ice-blocks as protection against them.

Ieuan The three crosses of the awkward *Frogger*-style beginning spoil the rest of the program. Excellent sound, especially the harmonious jingle. The graphics are also very good. *8/10*

Dilwyn The coloured snakes racing around the screen got on my nerves after a while, but technically it's a great effect — typical of many of the fine finishing touches to this game. *8/10*

Brian The second stage of the game is rather faster than the first, but great once you get to it. Lovely smooth animation and it's compatible with most joystick interfaces. *9/10*



OLIGOPOLY
CCS/£5.00

A business management simulation game in which companies have to be run by one to six players; they have to show a suitable degree of business acumen.

Brian *Oligopoly* is a very complex game which seems to succeed as an excellent business simulation. Sometimes it says 'Press any key', waits for a while and then goes ahead without a keypress — which is annoying because it's difficult enough to show a profit without anything else going wrong. *7/10*

Ieuan As a business program, it doesn't sell itself very well and seems as though it was originally written for the ZX81. It would probably benefit from a splash of colour and a new character set. *6/10*

Dilwyn It has a facility to use the printer if desired and the computer can run a rival firm. Perhaps there could be some educational uses here. *6/10*



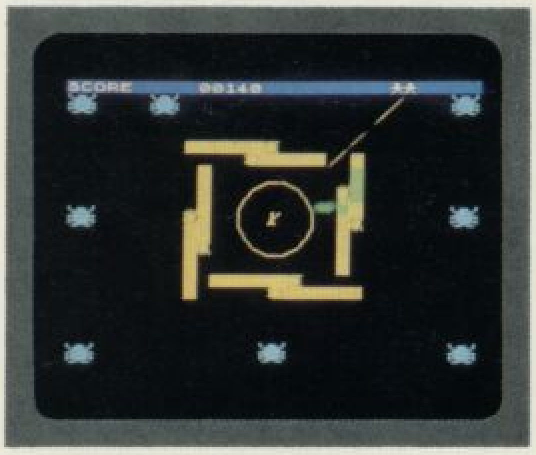
BEAR BOVVER
Artic Computing/£6.95

Ted the bear has to fetch batteries for his (Sinclair Research?) electric car — but is hindered by several 'teddy boys' who insist on making things difficult.

Dilwyn Fantastic graphics and animation accompanied by the 'Teddy Bear's Picnic' tune — which can be turned off once it gets to you. The only problem is the marring of the colour resolution where two colours meet. *9/10*

Ieuan Quite addictive and wonderful animation considering all the action involved. *Bear Bovver* uses a poster-style set of letters which is quite attractive. *9/10*

Brian Not too bad. Well thought-out, with the bonus drinks and the fact you're able to deter the opposition by dropping time bombs or burying them under a battery! Good sleeve notes are an advantage. *8/10*



DEVIL RIDES IN
Carnell Software/£5.95

You are a wizard in your magic circle and you have to use your magic talisman to do battle with the armies of Hell who throw spells at you. Neither you nor the attackers are allowed to touch the magic circle, even if continuous fighting has partly erased it.

Ieuan Reminds me a bit of *Asteroids*. The 'rotate through 45 degrees' feature is a nuisance — it should allow simultaneous keypress or joystick diagonals instead. *7/10*

Dilwyn Other than Ieuan's comment, a nice game. Slow at first, but once it gets going it's fun to play. *8/10*

Gerralt All events are covered with a 'zap' or 'ping'. The sound slows things down a bit which, once the game has got into top gear, causes a little jerkiness. There doesn't seem to be a time limit as long as you keep alive. *7/10*



RIDER
Virgin Games/£5.95

An MI5 agent parachutes into enemy territory, collects a Resistance motor bike and checks out the mined roads ready for the invasion — ta-rah and toodle-pip!

Brian Good colour and graphics throughout. The game also improves in speed after you've managed to get over the first stage — watch out for the sound effects too. *7/10*

Gerralt The parachute bit at the beginning is too slow and long, especially when you consider this game's meant to be about riding a bike — which, by the way, is by far the most interesting bit. *6/10*

Dilwyn This one really wasn't my cup of tea at all. I think it was the long intro that put me off the most. There's some you love and some you don't. Say no more. *4/10*



CAR JOURNEY
Heinemann/£9.95

Another educational game for eight-to-12 year olds which this time has a mathematical slant — towards teaching arithmetic, graphs and strategy planning. The idea is to travel around Britain choosing the appropriate vehicle and route for various tasks.

Dilwyn Fairly good graphics and route map. All units are metric for school use. It's a pity that the reference to the Welsh 'Eisteddfod' had to be mis-spelt though! *7/10*

Ieuan No sound again! Useful activity booklet and keyboard overlay. The package is better than most educational software, but I'm not sure how long it will keep the kids' interest. *6/10*

Gerralt Basically, a fairly simple concept done quite well; a bit more animation would probably have helped to keep the attention of the users. *7/10*

DUMPS OF DISTINCTION

Frustrated at producing tiny screen dumps on your* full-sized *printer? So was* Andrew Pennell *— until he developed this clever little machine code utility, that is!







If you're fortunate (ie. rich) enough to have an Epson RX80 or FX80 connected to your Spectrum, then this is *the* screen dump program for you.

Most printer interfaces are supplied with a high resolution dump routine, which produces a ZX Printer-sized copy of the screen. However, the copies lack the different colours of the original, and the colour system on the Spectrum can produce weird results when copied because only the screen bit-map is examined, and the colour information is ignored.

Nothing's ignored in this colour screen dump. However, not having the wherewithal for a full seven-colour printer, I decided to write a screen dump routine for my Epson that printed a large that printed a large picture with different shades simulating different pixel colours; the following program is the result, producing an 11-by 6-inch copy.

The program works on either the 16K or 48K Spectrum, as it moves RAMTOP down to address 32349; 48K owners cannot use their extra 32K of memory. For those with the inclination, the routine can be modified to run from address 65118, by changing all '205,20,127' sequences to '205,20,255', and by changing '33,244,126' to '33,244,254'.

Listing 1 shows the Basic loader for those with the Hilderbay interface; note that you must load the 'mini-software' first. For those with a Kempston interface, add the lines in Listing 2 and, for those with a Kempston Kempston E interface, add the lines in Listing 3. If you've got none of these interfaces, you'll just have to modify the machine code yourself — details to follow. And equally important, if your printer requires Line Feeds after Carriage Returns, add the line:

```
230 POKE 32478, 205: POKE 32479,20:
POKE 32480,127
```

Once you've entered and RUN the loader, save the routine to tape with SAVE "col copy"CODE 32350,250 — and the Basic loader after it, in case of a crash. Then load a SCREEN$, and type RANDOMIZE USR 32350. If your printer starts printing, you've probably got the code right. If it doesn't, or if the system crashes, then there must be a mistake and you should re-load the Basic and check it against the listings.

I think the best screen dumps are

those from the Ultimate programs, and some of these adorn this article. Unfortunately, they all contain a lot of black, so you'll need a new(ish) ribbon for best results. If you're using fan-fold paper 11-inches in length, position the perforations just above the print head before the dump — otherwise the next perforation

Listing 1

```
 100 CLEAR 32349
 110 RESTORE 1000
 119 REM POKE the setup routine
 120 FOR I=32350 TO 32369
 130 READ A: IF A=-1 THEN GO TO
150
 140 POKE I,A: NEXT I
 149 REM POKE the routine itself
 150 LET C=0
 155 FOR I=32370 TO 32531
 160 READ A: POKE I,A
 170 LET C=C+A
 180 NEXT I
 185 IF C<>13071 THEN PRINT "DAT
A ERROR": STOP
 190 FOR I=32532 TO 32599
 200 READ A: IF A=-1, THEN GO TO
220
 210 POKE I,A: NEXT I
 220 PRINT "FINISHED."
1000 DATA 205,0,91,24,15,-1
1100 DATA 62,27,205,20,127,62,65
,205,20
1110 DATA 127,62,3,205,20,127,14
,0,62
1120 DATA 27,205,20,127,62,42,20
5,20,127
1130 DATA 62,4,205,20,127,62,16,
205,20
1140 DATA 127,62,2,205,20,127,6,
0,197-
1150 DATA 205,170,34,71,4,62,1,1
5,16
1160 DATA 253,166,8,124,15,15,15
,230,3
1170 DATA 246,88,103,70,8,120,32
,3,15
1180 DATA 15,15,230,7,33,244,126
,135,135
1190 DATA 95,22,0,25,6,3,126,205
,20
1200 DATA 127,35,16,249,193,4,12
0,254,176
1210 DATA 56,199,62,13,205,20,12
7,62,10
1220 DATA 0,0,0,12,32,159,62,27,
205
1230 DATA 20,127,62,65,205,20,12
7,62,12
1240 DATA 205,20,127,201.224,224
,224,0,192
1250 DATA 96,192,0,160,64,160,0,
32,64
1260 DATA 128,0,96,0,96,0,64,0,6
4
1270 DATA 0,0,64,0,0,0,0,0,0
1300 DATA 197,229,205,254,91,225
,193,201,-1
```

The Basic loader for use with a Hilderbay interface.

```
1000 DATA 1,191,227,62,129,237,1
21,62,15
101' DATA 237,121,24,7,-1
1300 DATA 243,197,1,191,226,30,1
4,237,80
1310 DATA 203,66,32,250,6,224,23
7,121,6
1320 DATA 227,237,89,28,237,89,2
51,205,84
1330 DATA 31,210,0,13,193,201,-1
```

Add these lines to the loader to use with a Kempston interface.

```
1000 DATA 62,3,205,1,22,24,13,-1
1300 DATA 245,62,27,215,241,215,
201,-1
```

Add these lines to the loader to use with a Kempston E interface.

Listing 4

This listing is taken from the Hisoft GENS assembler and uses # to denote hex numbers and % for binary numbers. It also supports conditional assembly.

```
                  100 ; EPSON COLOUR COPY
                  110 ; for RX/FX80s
                  120 ; (C) A.PENNELL 1984
                  130 ;define interface types
0001              140 HILDER EQU  %1
0002              150 KEMPN  EQU  %10
0004              160 KEMPE  EQU  %100
0001              170 TYPE   EQU  HILDER ;change to suit interface
7E5E              180 START  EQU  32350
7E5E              190        ORG  START
                  200 ;Hilderbay set-up routine
0001              210        IF   TYPE&HILDER
7E5E CD005B       220        CALL 23296 ;setup routine
7E61 180F         230        JR   BEGIN
7E63              240        END  ;Hilderbay setup
                  250 ;KEMP normal setup
0000              260        IF   TYPE&KEMPN
7E63              270        LD   BC,#E3BF
7E63              280        LD   A,129
7E63              290        OUT  (C),A
7E63              300        LD   A,15
7E63              310        OUT  (C),A
7E63              320        JR   BEGIN
7E63              330        END
                  340 ; KEMPSTON E setup
0000              350        IF   TYPE&KEMPE
7E63              360        LD   A,3
7E63              370        CALL #1601 ; open "P"
7E63              380        JR   BEGIN
7E63              390        END
001B              400 ESC    EQU  27
                  410 ;
                  420 ; the routine itself
7E72              430        ORG  START+20
7E72 3E1B         440 BEGIN  LD   A,ESC
7E74 CD147F       450        CALL OUTCH
7E77 3E41         460        LD   A,"A"
7E79 CD147F       470        CALL OUTCH
7E7C 3E03         480        LD   A,3
7E7E CD147F       490        CALL OUTCH ;set up small line feeds
7E81 0E00         500        LD   C,0 ;zero X counter
7E83 3E1B         510 NLINE  LD   A,ESC
7E85 CD147F       520        CALL OUTCH
7E88 3E2A         530        LD   A,"*"
7E8A CD147F       540        CALL OUTCH
7E8D 3E04         550        LD   A,4 ;put in mode 4
7E8F CD147F       560        CALL OUTCH
7E92 3E10         570        LD   A,16
7E94 CD147F       580        CALL OUTCH
7E97 3E02         590        LD   A,2
7E99 CD147F       600        CALL OUTCH ;set up for 3*176 bits of data
7E9C 0600         610        LD   B,0 ;zero Y counter
7E9E C5           620 NXY    PUSH BC
7E9F CDAA22       630        CALL #22AA ;HL=screen memory
7EA2 47           640        LD   B,A
7EA3 04           650        INC  B
7EA4 3E01         660        LD   A,1
7EA6 0F           670 L1     RRCA
7EA7 10FD         680        DJNZ L1
7EA9 A6           690        AND  (HL) ;Z if ink,NZ if paper
7EAA 08           700        EX   AF,AF'
7EAB 7C           710        LD   A,H
7EAC 0F           720        RRCA
7EAD 0F           730        RRCA
7EAE 0F           740        RRCA
7EAF E603         750        AND  3
7EB1 F658         760        OR   #58
7EB3 67           770        LD   H,A ;HL=attribute byte
7EB4 46           780        LD   B,(HL) ;B=ATTR
7EB5 08           790        EX   AF,AF'
7EB6 78           800        LD   A,B ;A=ATTR
7EB7 2003         810        JR   NZ,INK
7EB9 0F           820        RRCA ;if PAPER then /8
7EBA 0F           830        RRCA
7EBB 0F           840        RRCA
7EBC E607         850 INK    AND  7 ;mask other bits
7EBE 21F47E       860        LD   HL,TABLE
7EC1 87           870        ADD  A,A
7EC2 87           880        ADD  A,A
7EC3 5F           890        LD   E,A
7EC4 1600         900        LD   D,0
7EC6 19           910        ADD  HL,DE ;HL=data        specified colour
7EC7 0603         920        LD   B,3 ;=no of bytes per pixel
7EC9 7E           930 OUTLP  LD   A,(HL) ;read byte
7ECA CD147F       940        CALL OUTCH ;send it
```

will appear in the middle of the dump. if you break in the middle of it, you may find that your printer acts strangely, because it's in bit mode; the best way to clear it is by switching the printer off, then on again.

MODUS OPERANDI

The screen is copied sideways, and consists of a 176 x 256 grid of pixels. If each screen pixel is represented by a three-by-three grid on the printer, then this points to a printer resolution of 528 x 768. The former figure rules out the old Epsons (MX-type and assorted look-alikes) with their resolution of only 480 dots per line. The new Epsons have a resolution of 640 dots per line in bit image mode 4, and that's why we're using them here. As well as this, the vertical Line Feed distance has to be changed to three dots, which is 3/72 of an inch, and is changed with the ESC "A" command.

CRACKING THE CODE

The full assembler listing is shown in Listing 4, printed from the Hisoft *GENS* assembler, which uses '#' to denote hex numbers, and '%' for binary numbers. It also supports conditional assembly, which I have used to determine the set-up and output routines for different interfaces. If your assembler does not support conditional assembly, just leave out all mnemonics that refer to interfaces other than your own. If you have a different interface, substitute your own set-up routine at START, and your own output routine at OUTCH. Note that OUTCH must preserve the values of HL and BC.

The copy routine proper starts at BEGIN, which firstly resets the Line Feed distance to 3/72 of an inch and zeros C — which is used as a counter for the X co-ordinate. Next the printer is put into graphics mode 4, and the B reg (which is the Y counter) is set to zero. The ROM routine at 22AA is called, which calculates screen location HL and bit position B from the screen co-ordinates in B and C, and the pixel colour calculated from the attribute file. The relevent data is read from TABLE and printed out, and this is repeated for all 176 Y positions. After this, a Newline is sent (and optionally a Line Feed), and all 256 X positions are done. Finally, the Line Feed distance is reset to 12/72 of an inch. The data at TABLE contains the bit patterns for each colour. They were originally chosen quite arbritarily, but I left them as they seem to produce good results. If you don't like the shades produced, feel free to change the data, though note that the fourth byte for each colour should be zero. Also, beware, Listing 4 was made with the Hilderbay interface selected, so no object code was generated for either of the Kempston interfaces. YS

```
7ECD 23          950          INC  HL
7ECE 10F9        960          DJNZ OUTLP ;do 3 bytes
7ED0 C1          970          POP  BC ;restore X & Y
7ED1 04          980          INC  B
7ED2 78          990          LD   A,B
7ED3 FEB0       1000          CP   176
7ED5 38C7       1010          JR   C,NXY ;do all 176
7ED7 3E0D       1020          LD   A,13 ;end of line so do a CR
7ED9 CD147F     1030          CALL OUTCH
7EDC 3E0A       1040          LD   A,10 ;LF code
7EDE 00         1050          NOP  ;room for CALL OUTCH
7EDF 00         1060          NOP
7EE0 00         1070          NOP
7EE1 0C         1080          INC  C
7EE2 209F       1090          JR   NZ,NLINE ;do all 256 X pixels
7EE4 3E1B       1100          LD   A,ESC ;reset line feed distance
7EE6 CD147F     1110          CALL OUTCH
7EE9 3E41       1120          LD   A,"A"
7EEB CD147F     1130          CALL OUTCH
7EEE 3E0C       1140          LD   A,12
7EF0 CD147F     1150          CALL OUTCH
7EF3 C9         1160          RET  ;finish
                1170 ;data table for colour patterns
7EF4 E0         1180 TABLE    DEFB %11100000 ;black
7EF5 E0         1190          DEFB %11100000
7EF6 E0         1200          DEFB %11100000
7EF7 00         1210          DEFB 0
7EF8 C0         1220          DEFB %11000000 ;blue
7EF9 60         1230          DEFB %01100000
7EFA C0         1240          DEFB %11000000
7EFB 00         1250          DEFB 0
7EFC A0         1260          DEFB %10100000 ;red
7EFD 40         1270          DEFB %01000000
7EFE A0         1280          DEFB %10100000
7EFF 00         1290          DEFB 0
7F00 20         1300          DEFB %00100000 ;magenta
7F01 40         1310          DEFB %01000000
7F02 80         1320          DEFB %10000000
7F03 00         1330          DEFB 0
7F04 60         1340          DEFB %01100000 ;green
7F05 00         1350          DEFB %00000000
7F06 60         1360          DEFB %01100000
7F07 00         1370          DEFB 0
7F08 40         1380          DEFB %01000000 ;cyan
7F09 00         1390          DEFB %00000000
7F0A 40         1400          DEFB %01000000
7F0B 00         1410          DEFB 0
7F0C 00         1420          DEFB %00000000 ;yellow
7F0D 40         1430          DEFB %01000000
7F0E 00         1440          DEFB %00000000
7F0F 00         1450          DEFB 0
7F10 00         1460          DEFB %00000000 ;white
7F11 00         1470          DEFB %00000000
7F12 00         1480          DEFB %00000000
7F13 00         1490          DEFB 0
7F14            1500          END
                1510 ; OUTCH - the send character to printer routine
7F14            1520 OUTCH    EQU  $
                1530 ; Hilderbay O/P routine
0001            1540          IF   TYPE&HILDER
7F14 C5         1550          PUSH BC
7F15 E5         1560          PUSH HL ;save regs
7F16 CDFE5B     1570          CALL 23550 ;call the routine
7F19 E1         1580          POP  HL
7F1A C1         1590          POP  BC ;restore regs
7F1B C9         1600          RET
7F1C            1610          END
                1620 ; KEMPSTON normal output
0000            1630          IF   TYPE&KEMPN
7F1C            1640          DI
7F1C            1650          PUSH BC
7F1C            1660          LD   BC,#E2BF
7F1C            1670          LD   E,14
7F1C            1680 KBUSY    IN   D,(C)
7F1C            1690          BIT  0,D
7F1C            1700          JR   NZ,KBUSY
7F1C            1710          LD   B,224
7F1C            1720          OUT  (C),A
7F1C            1730          LD   B,227
7F1C            1740          OUT  (C),E
7F1C            1750          INC  E
7F1C            1760          OUT  (C),E
7F1C            1770          EI
7F1C            1780          CALL #1F54 ;test break
7F1C            1790          JP   NC,#0D00 ;error if pressed
7F1C            1800          POP  BC
7F1C            1810          RET
7F1C            1820          END
                1830 ; KEPMSTON E output
0000            1840          IF   TYPE&KEMPE
7F1C            1850          PUSH AF
7F1C            1860          LD   A,ESC
7F1C            1870          RST  #10
7F1C            1880          POP  AF
7F1C            1890          RST  #10
7F1C            1900          RET
7F1C            1910          END
```

You could be forgiven for thinking that producing a successful computer game depends merely upon writing any old piece of code, slapping it on to a tape, packaging it in an attractive box and making exaggerated claims in the advertisements — especially when you see so many *Space Invader, Frogger* and *Pacman* clones still being written. However, Mr Chip Software rather likes introducing the odd note of variety.

Some time ago, it released a sort of government management strategy game called *Westminster*, where the player takes on the role of Prime Minister (not an enviable position) and tries to run the country by juggling with the economy and dealing with the various opposition groups. It was written originally for the Commodore 64, but so pleased are they all with the finished product at Mr Chip's factory, the news is they've decided to re-write it for the Spectrum.

But all is not well! According to Mr Chip's Doug Brainsby, "although the game got a good response from both the public and the press, the distributors were unwilling to take it. And if their minds remain unchanged, this could lead to an even bigger problem now that we've decided to go for the (potentially) much larger Spectrum market." Distributor reactions are awaited with interest.

WRATH OF PUNTERS

The latest word from Carnell Software concerns the long-awaited *Wrath of Magra* adventure, forming part of a series that includes *Volcanic Dungeon* and *Black Crystal.* It was announced some time ago as "coming soon", but it's still not appeared in the shops. Carnell's Simon Bassett says, "the program's going through its final testing stage at the moment, but this is causing me all sorts of problems. It's so complex and difficult to play that it takes ages to work out exactly what's going on". So hang on in there — it should be available over the next few weeks.

As well as trying to get to grips with the *Wrath of Magra*, Carnell is also re-vamping *Volcanic Dungeon*, so that it'll now include high resolution graphics. And what's more, there'll be the added advantage (?) of its being compatible with Currah's Micro-Speech unit, allowing you the dubious pleasure of being able to sit back and listen to the game's vocabulary of over 100 words in electronic tones.

FLYING HIGH

Moving on, Durrel Software's *Harrier Attack* program, although successful, apparently received a mixed reaction from the punters. A spokesperson said, "we had people complaining about it, because of the weapons, the violence and all that shooting. So, in our latest game, *Scuba Dive (as featured in last issue. Ed.)* we didn't even allow the odd harpoon to be fired. But still people complained." Anyway, ignoring the criticism, Durrel has decided to revert to its original theme and is currently planning a follow-up to *Harrier Attack*. In fact, to get it *really* realistic, the company is actually sending off a team of programmers to Westmoreland so they can get a feel of flying before sitting down to write their sequel.

HERE COME DE JUDGE?

In an effort to come up with original ideas for new computer games, some software houses are turning to familiar comic strip characters and giving them a new angle. This is just what Channel 8 Software is planning to do, anyway.

Spokesperson, John Williams, tells us that "negotiations with IPC Magazines are currently in progress. But I can't go into details at the moment as this could affect the deal." However, Mr Williams was able to confirm that Channel 8 has an idea for a program, and it's quite clear this requires the use of one of said IPC's characters. Perhaps negotiations are faltering for he added, somewhat defiantly, "the program will still appear, even if the talks turn out to be fruitless. But that'll mean it will have to take on a slightly different form".

Could this be the start of a whole new area of computer games software?

SWEET NOTHINGS

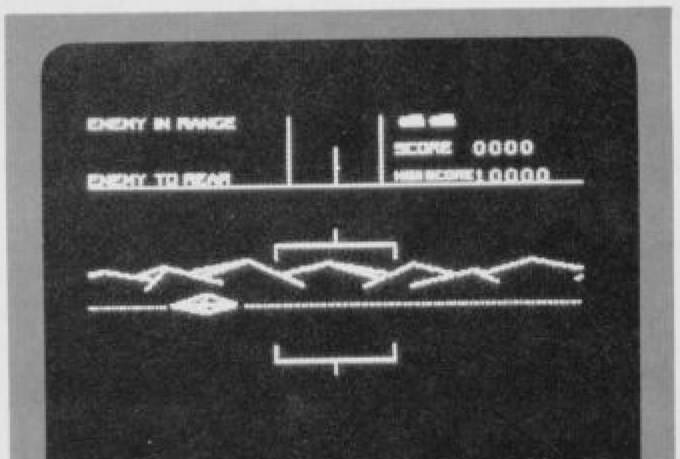
Quicksilva is also kicking around ideas for using fictional characters but, as usual, the best that Mark Eyles could come up with was the boring "nothing's been signed yet, and it could be some time before we reach that stage". He was equally cagey about a follow-up to the company's highly successful and original *Ant Attack* — you know, player pitted against giant ants in an attempt to save mate, etc, etc. Mark does say that "Sandy White (the *Ant Attacker* himself) is coming to see us soon, so it's quite likely that a new program will be discussed." Well, well, well — what a surprise! Will the nice Mr Eyles kindly 'open up' a little more in the future . . . 'Ve haf vays ov making you tok' you know.

OUT OF THE PIT

Rabbit Software is planning a new game that will feature a fairly unlikely collection of enemies. Provisionally entitled *The Pit*, it's quite likely it will be called something like *Vortex* once it gets out of the testing stage.

According to Rabbit director, Terry Grant, "the game is set in a cavern with the player taking control of a ship". Among the obstacles to overcome will be "witches, bunnies, ghosts, televisions and funny white blobs — but I don't know quite what they are." Neither did Mr Grant seem to know the exact aim of the game, but he was sure that it was very good.

But whatever its final qualities, Rabbit is planning the added attraction of giving away a free blank tape with each copy of *The Pit* (or *Vortex*, or whatever it is going to be called). This might, of course, be quite a convenient carrot to dangle since the company's just acquired a tape duplication machine — which probably means it's got a cheap, bulk-buying deal worked out for raw tape.

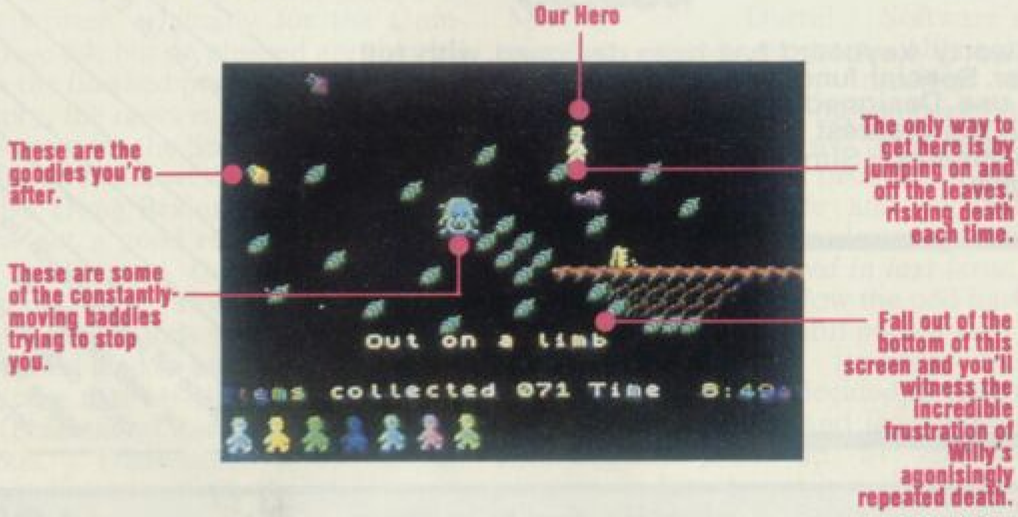


Battle Zone is an arcade-style game currently being evaluated by Quicksilva. Basically, you just have to blast those enemy tanks off the face of the Earth.

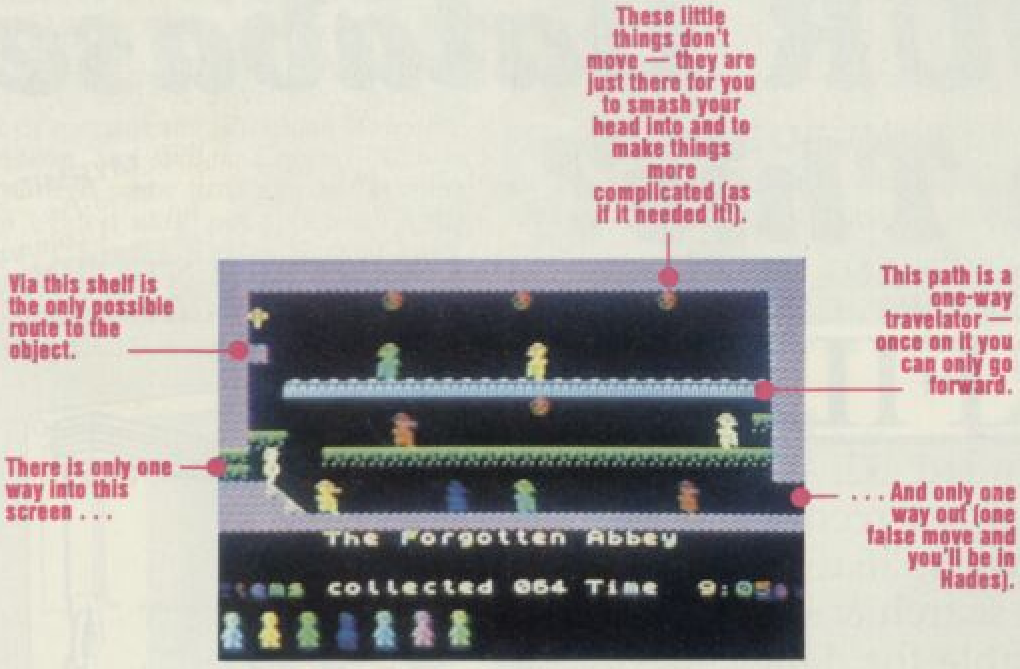
JOIN THE JET SET

Keeping one jump ahead of the software market seems to come naturally to author of Jet Set Willy, Matthew Smith. Sue Denham checks out his follow-up to Manic Miner, where Miner Willy hits the big time.

OUT ON A LIMB



THE FORGOTTEN ABBEY



Succeeding with *Jet Set Willy* demands a number of basic skills as well as perverse dedication. The first problem is to discover how to get to each room, and once there, how to get to each object — we've solved most of that for you here. Certain screens, such as 'Out on a limb' pictured here, demand extreme accuracy with the keyboard and although the game is also configured for joysticks they are not accurate enough as often you need to jump from what appears to be the last possible pixel to gain the goodies. Others such as 'The Forgotten Abbey' demand nano-second timing and synchronisation of jumps. This room might look intimidating here — but wait till you see it with the Napoleonic ducks in action, particularly when you realise that you have to leap over every one of them! Most rooms (see over) demand both skills simultaneously.

Every now and then, there comes a program that somehow prevents reviews from being written in a hurry —simply because tapping typewriter keys is cold comfort after you've negotiated the perilous journey up The Megatrunk, or collected the goodies from The Forgotten Abbey, or entered The Chapel and lived, or . . . . .

Star of the Speccy screen, Matthew Smith, has finally delivered his follow-up to *Manic Miner*, and it's every bit as good and refreshing as the original. The story line is as weak as ever — some nonsense about Willy having thrown a party and the guests having left lots of champagne glasses strewn about his mansion (altered slightly in the production version). The aim is for him to collect all of these, because his housekeeper won't let him into the bedroom until he has. Weak it may be, still 'it's the game itself wot matters'.

The game is colourful, fast and ingenious. The controls are simple; you can move left or right and you can jump effortlessly into the air. And that's all there is to it — except, of course, that this is where all your problems begin!

CLOCK THIS

Press Enter and you're whisked from the title page to your first glimpse of Willy's 60-room mansion. That's you standing in the bath, staring at a flashing tap and a toilet (complete with moving seat (a la *Manic Miner*). The moral of the game is that virtually everything that isn't flashing will kill you should you be foolish enough to walk into it. You've guessed it . . . the flashing objects are the ones you have to collect; there are 83 of them in all and the majority are very difficult to find indeed.

At the bottom of the screen, there's an indication of time. You begin your quest at seven in the morning (it must have been one heck of a party!) and the idea is to get into your bedroom by the hour of midnight. No, that doesn't mean you'll be sitting at the keyboard for 17 hours (although that wouldn't surprise me) for Matthew has thoughtfully shortened each minute to around 40 seconds. But anyway, this is unlikely to bother users for quite a time. . . in the many weeks this game has taken to review, the clock has still never made it to eight in the morning!

When you begin playing you start with eight lives, which at first seems a bit excessive (ho ho) — until you venture past The Bathroom. It's worthwhile just having a wander around to get a feel for the way Miner Willy handles; for example, you can get Willy to hang precariously to a surface by what looks like a single pixel before making that important leap — and in some cases that's exactly what you'll have to do.

BATHTIME'S OVER

One step out of The Bathroom and you're thrust into the thick of Matthew Smith's fertile imagination — and what a place that must be! You can forget all

about malevolent space invaders and greedy Pac-persons, here the baddies are Swiss Army knives, razor blades, mini-chefs, grotesque faces, wobbling jellies, rolling eggs, ballet-dancing gerbils, a Monty Python foot, and . . . need I go on?

That's not all you have to watch out for. The program has a nasty habit of thrusting you on-screen in a room you've just lost a life in — leaving you powerless to prevent all the remaining lives being eaten up in the same way. Try making some of the leaps across The Orangery, for instance. Should you miss your footing you're likely to end up falling down into The Swimming Pool and dying. The rest of your lives will then automatically be swallowed up in the same way, leaving you impotent with rage and uselessly hitting every key in sight, in a vain atttempt to prevent the inevitable.

Another strange quirk is the way in which the rooms have been laid out. The top floor seems to have more rooms than the floor below, and the basement has even fewer. For instance, if you go from The Wine Cellar into the next room, you'll end up in The Forgotten Abbey — which according to my calculations is right over the other side of the mansion!

You can also reach some strange places by getting to the highest point of some rooms and jumping up. The first

JSW-A HACKER'S GUIDE

***If you always type MERGE "" whenever you load a game for the first time, then you can count yourself amongst that select programming group known as 'hackers'. Join* Andrew Pennell *on a journey through the machine code magic that comprises* Jet Set Willy.**

Although playing any game of the quality of *Jet Set Willy* is in itself great fun, the more mischievous among us get a double helping of kicks by peering into sections of the program — both to examine its structure and to alter certain attributes (in other words, to cheat).

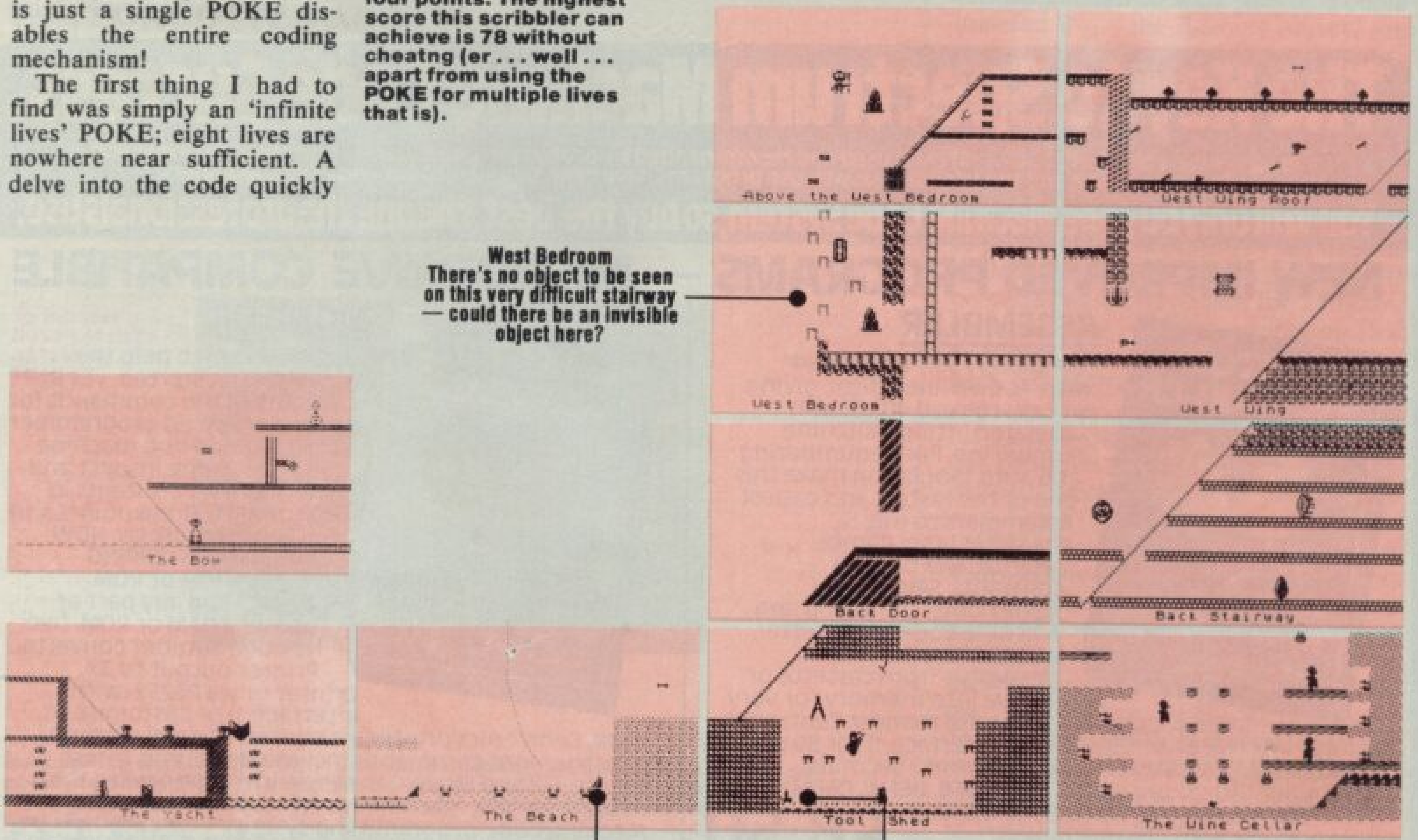
The program itself can hardly be described as fully protected. Although the colour chart supplied will defeat most home pirates, the keener and more able ones will resort to any lengths to get bootlegs of the game. I know of one player who typed the whole chart into his wordprocessor, and another who dutifully duplicated it all with felt tips. This latter soul gave me the biggest laugh — the fact is just a single POKE disables the entire coding mechanism!

The first thing I had to find was simply an 'infinite lives' POKE; eight lives are nowhere near sufficient. A delve into the code quickly unearthed the important DEC (HL), which was duly NOP'd out. This, however, is not the perfect solution. You'll probably have discovered that once the Attic has been visited, the program irrevocably alters itself — from there on in it's instant death if you enter four particular rooms. With infinite lives you're much more likely to find the Attic and, once you do, most of the rooms become blocked permanently. The (rather crude) solution I employ is simply to power down and re-load *Willy* from Microdrive cartridge, although cassette users could have a problem here.

Incidently, my current version of *JSW* (on Microdrive, of course) has a menu on it complete with options to choose the number of lives, which screen to start on and the numbers of objects within the game. The code itself has also been modified so that by pressing a certain key, the screen contents can be saved on cassette as a SCREEN$ file and, using the routines described in my

There's no need to bother counting — there are 82 visible objects. Including the four in the conservatory. I also know of a hidden one and one that is worth four points. The highest score this scribbler can achieve is 78 without cheatng (er . . . well . . . apart from using the POKE for multiple lives that is).

Swimming Pool
Entering this room (from any direction) automatically credits you with another object.



West Bedroom
There's no object to be seen on this very difficult stairway — could there be an invisible object here?

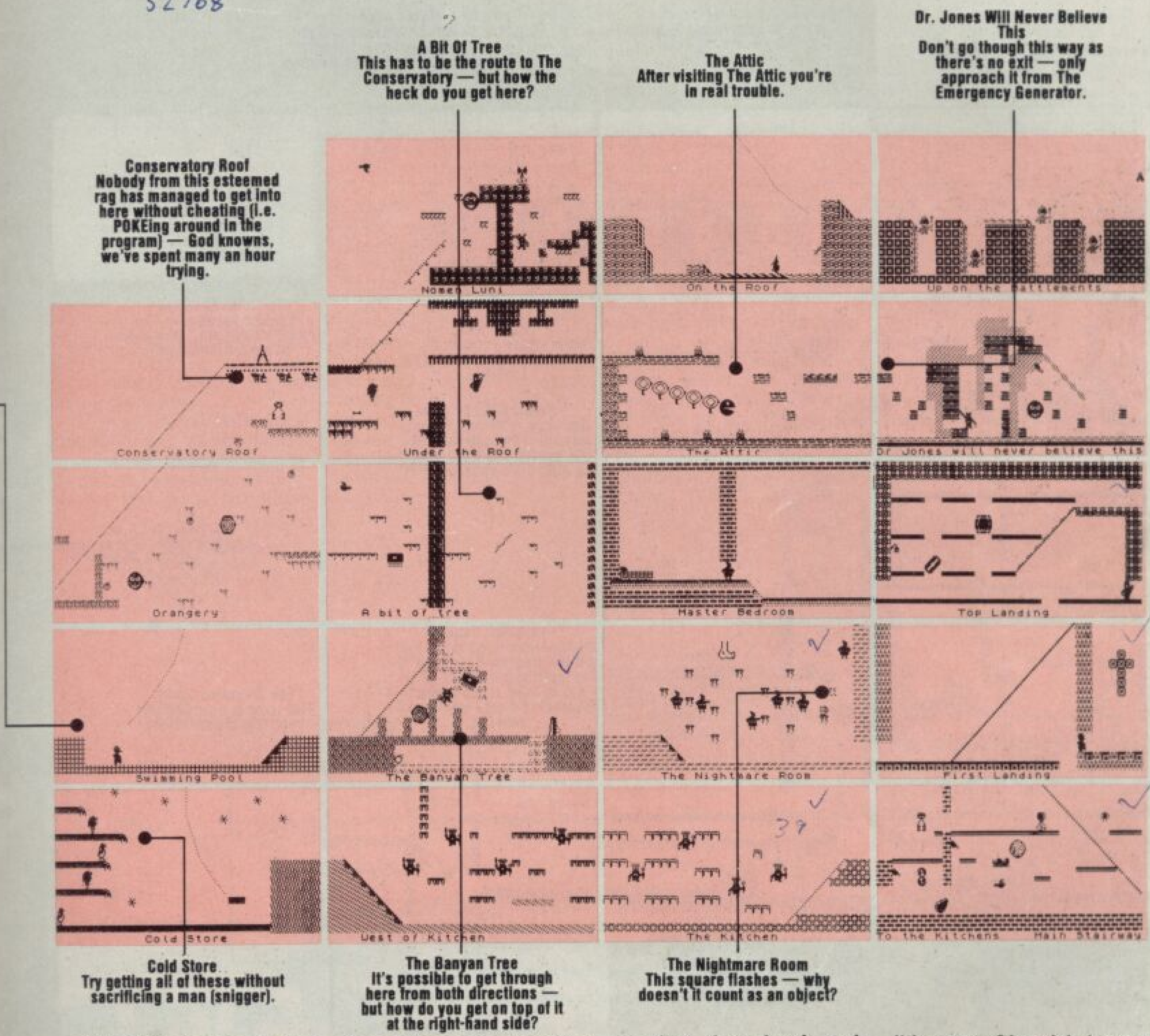
The Beach
This object is worth 2 points.

Tool Shed
This room presents a interesting dilemma — whether to sacrifice a man and gain the hammer the easy way — or try the hard way and die in the process.

experience of this came after a timely leap from Rescue Esmerelda, which sent poor Willy headlong into the floor of Ballroom East. Also, if you try jumping off most of the other rooms on the top floor, Willy ends up in The Off Licence. Whether this just shows Matthew Smith's alcoholic sense of humour, I'll probably never know, but after a few hours of *Jet Set Willy*, it turns out not to be a bad suggestion at all.

Unlike Andrew Pennell (the little cheat!), the approach in the *YS* office was simply one of striking out with eight meagre lives, in an attempt to find all the rooms. Having located around 45 of them, we seemed to come across a bug: each time we walked into certain rooms, Willy lost all his lives. It was time for a frantic phone call to Software Projects' Alan Morton. "Ahah', said Alan, "you didn't by any chance visit The Attic did you?" Sure 'nuff, we had — and very proud we were at the time. "Well, that's just a little something we put in to make it a bit more difficult", came the heavily understated reply. (The feeling our end is that it's a bug being turned into an asset — but who knows, we could be wrong!)

Indeed, it does make the game "a bit more difficult" — in fact, nigh on impossible to be precise. Once you visit The Attic, the four guardians from The Chapel race off to guard the entrances to The Kitchen, West of Kitchen, Cuc-



article *Print Routines* in this issue, dumped onto my Epson printer.

Anyway, — getting back to the Attic — I spent quite some time trying to find the piece of code that 'switched-in' this clever, if frustrating, effect; alas, to no avail. But the time wasn't wasted because during my investigations I turned up a number of other useful things including a way to find out the final solution. (By the way, if anyone out there has sorted out the POKE to disable the catastrophic 'Attic Attack' feature, please tell us rere at *YS*.)

The first thing that amazed me about *JSW* was the fact that there's only about 4K of machine code in the whole program! There's some 22K of data for the rooms, sprites and sound, while the remaining 12K or so is all used as 'workspace' by the other routines. In fact, much of the code is concerned with scanning the keyboard and all the possible types of joystick (except the Interface 2 type), not just to move Willy!

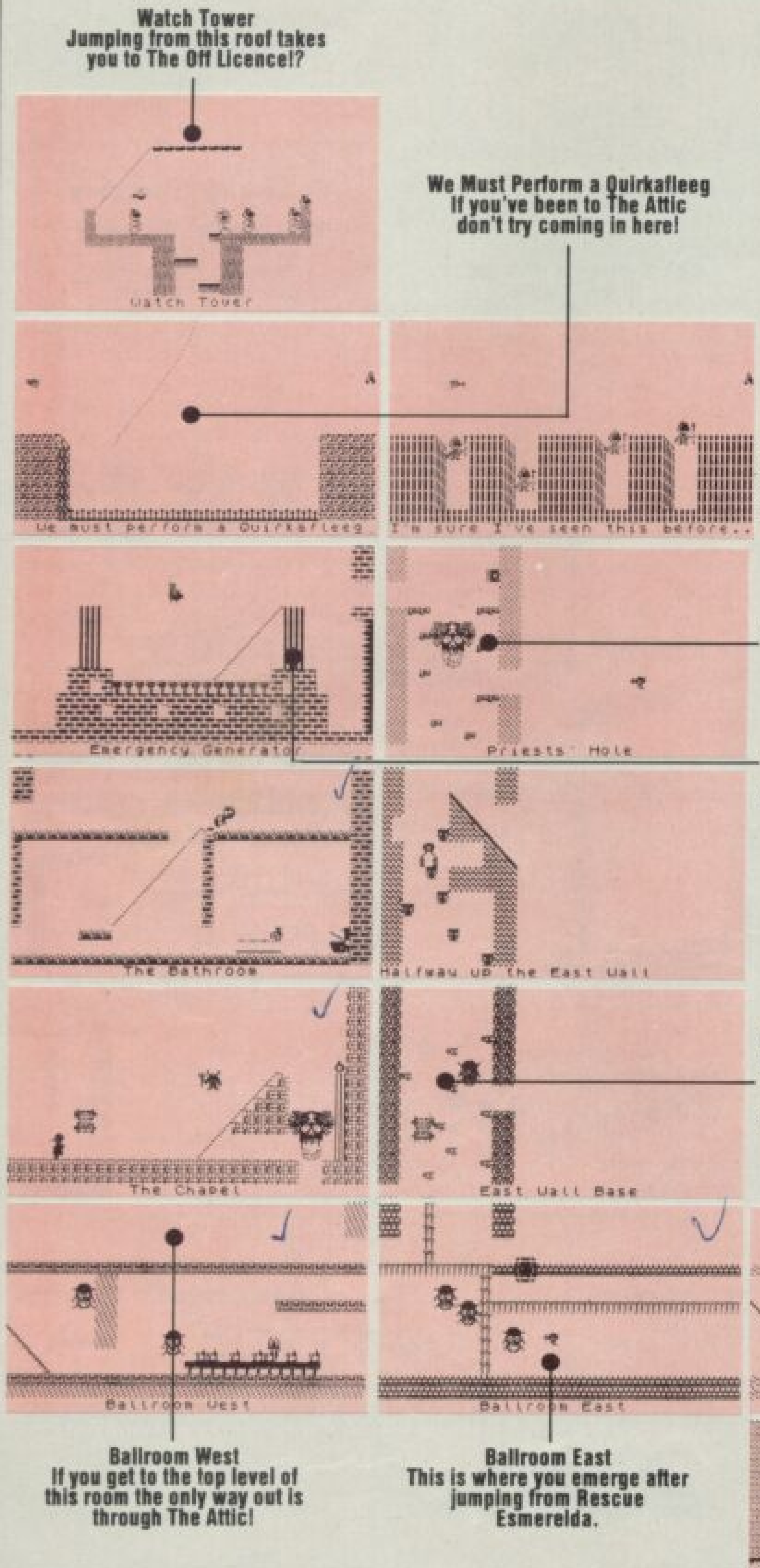
As you may have discovered on the original *Manic Miner*, you could get to any screen after typing in a nine-digit number (or on the Software Project's version, a 10-digit word) while playing the game. On *Jet Set*

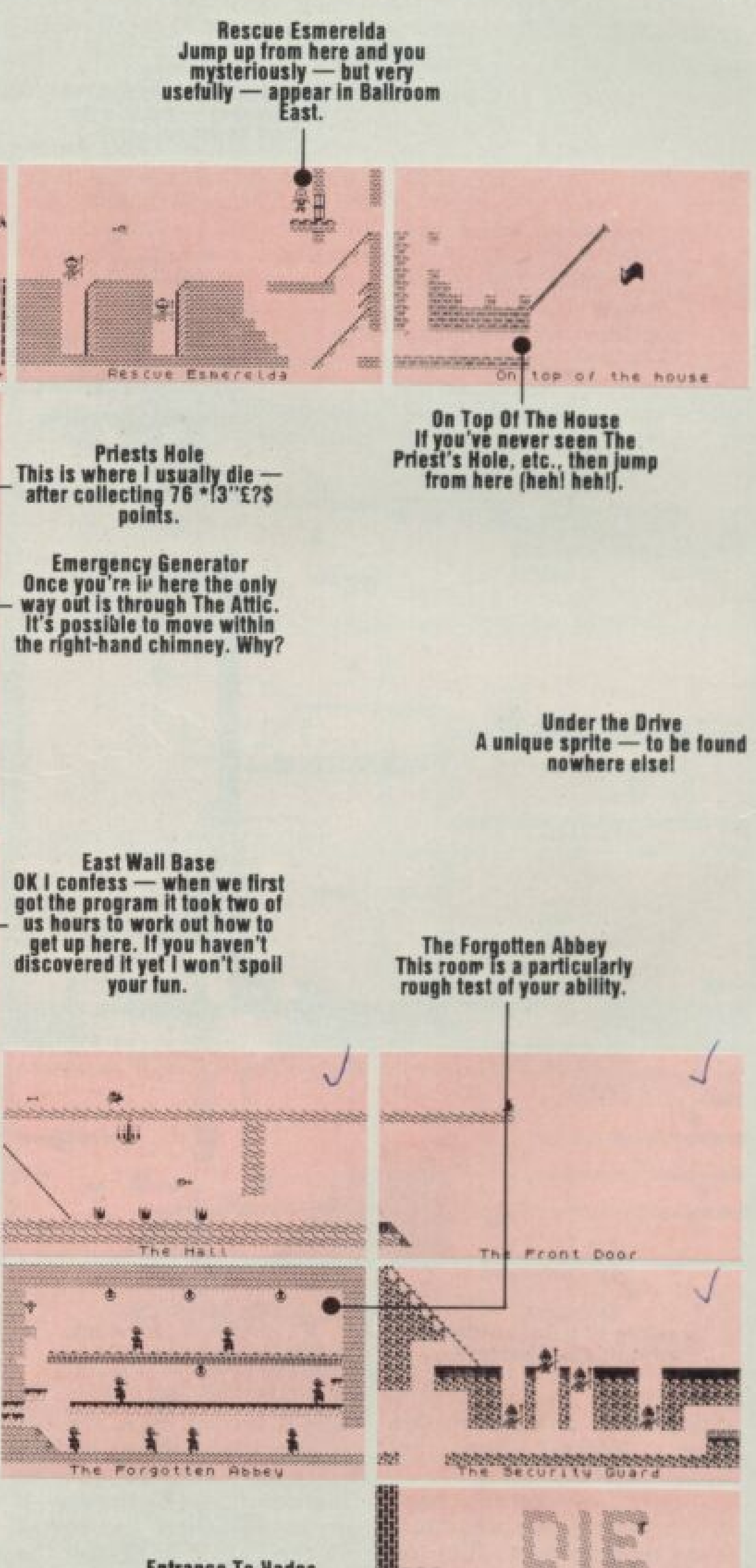
koo's Nest and East Wall Base. So, for goodness sake remember to check these places out first (and all rooms beyond) before you set foot in The Attic — otherwise you'll only have to re-load the program from tape again and start over.

Another clever little trick you discover, even before getting to indulge in the delights of the game, is the way Matthew Smith has chosen to 'anti-pirate' his program. Using a colour chart (don't lose it or you'll be in a mess), you have to type in a code of four colours which you access from the chart via co-ordinates flashed on-screen. Obviously it's not fool-proof, but it should slow 'em down a bit.

Like *Manic Miner*, *Jet Set Willy* has a charm which sets it aside from virtually every other game on the market. On a personal level, I find *Jet Set Willy* to be infinitely superior to its predecessor — if only because a practised Jet Setter can travel throughout the entire gamut of rooms without dying; my failure to complete the 12th level of *Manic Miner* prevented me from ever having to face the traumas of the following levels.

Most of the objects cached in *Jet Set Willy* are attainable, but there are some which, even when working from carefully scaled maps of each room, seem impossible to retrieve without sacrificing a life. For instance, there's a tricky one to get on the third level of the Cold





Willy it's a little bit harder. Here you have to be in a particular room at a particular height when you type out the 10 letters. But the problem is, having done this, you enter a room not at a 'safe' place, but at your current position — so extreme care must be taken when changing rooms using this method. You'll find you can only access 39 rooms like this anyway (answers on a postcard please for how to access the other 21).

The first person to finish *JSW* received a prize of Champagne and glasses from Software Projects, but to prove your *bone fides* you had first to tell the company how many hidden objects there were and what hap-

Store, a couple in the Wine Cellar, and the one in the Nightmare Room; this is made even more difficult by the sudden transformation of the Miner Willy character into an awkward flying pig shape. But if you really want to set yourself a task, try going after the goodies on the Conservatory Roof and see how well you make out.

If you enjoyed *Manic Miner*, then *Willy* is going to seem like the proverbial manna from heaven. Matthew Smith seems to have incorporated the best of his original creation, let none of his apparent fame spoil his wonderful sense of humour, and firmly set the blueprint for what I'm sure will be a very successful range of games — in much the same sort of way that Psion originally planned for Horace (remember him?).

In the meantime, it's good to see a program that'll rattle the software houses a bit and get them thinking along less traditional lines for their future releases. Full marks then to A&F Software for its *Chuckie Egg* which appeared in the wake of *Manic Miner*. Matthew Smith, meanwhile, is now happily ensconced in the Software Projects team (soon to be a director, we hear) — let's hope the association is both long and happy.

Okay, review over. Now, I think I'll just go and check out the Cuckoo's Nest — I almost managed to get the sparkler last time I tried. . . YS

pened when you then tried to go to bed. My first attempt at winning the splendid prize involved a couple of POKEs to remove the wall above Maria in the Master Bedroom (trying to get rid of Maria herself proved futile!). The POKEs worked and I could jump over her and into the bedroom proper. Touching the bed had no effect, while touching the pillow proved fatal — this was obviously not the way to cheat. A bit more time misspent listing out the code revealed both answers — and a single POKE allows anyone to see the final graphic effect, having picked up just one object! I'm sure you'll be happy to know that I came nowhere near the first to come up with the final answer though — no champers for me . . .

Anyway, I next built up a complete map of the house by sticking together screen dumps of each room — in a similar fashion to the illustration included with this article. However, I was still unable to see all of the objects. Finally, after much searching I stumbled upon a table that held the locations of each object — and found a few surprises. Some of them count as two, and others are 'invisible'; indeed, there's one somewhere on the First Landing which still eludes me.

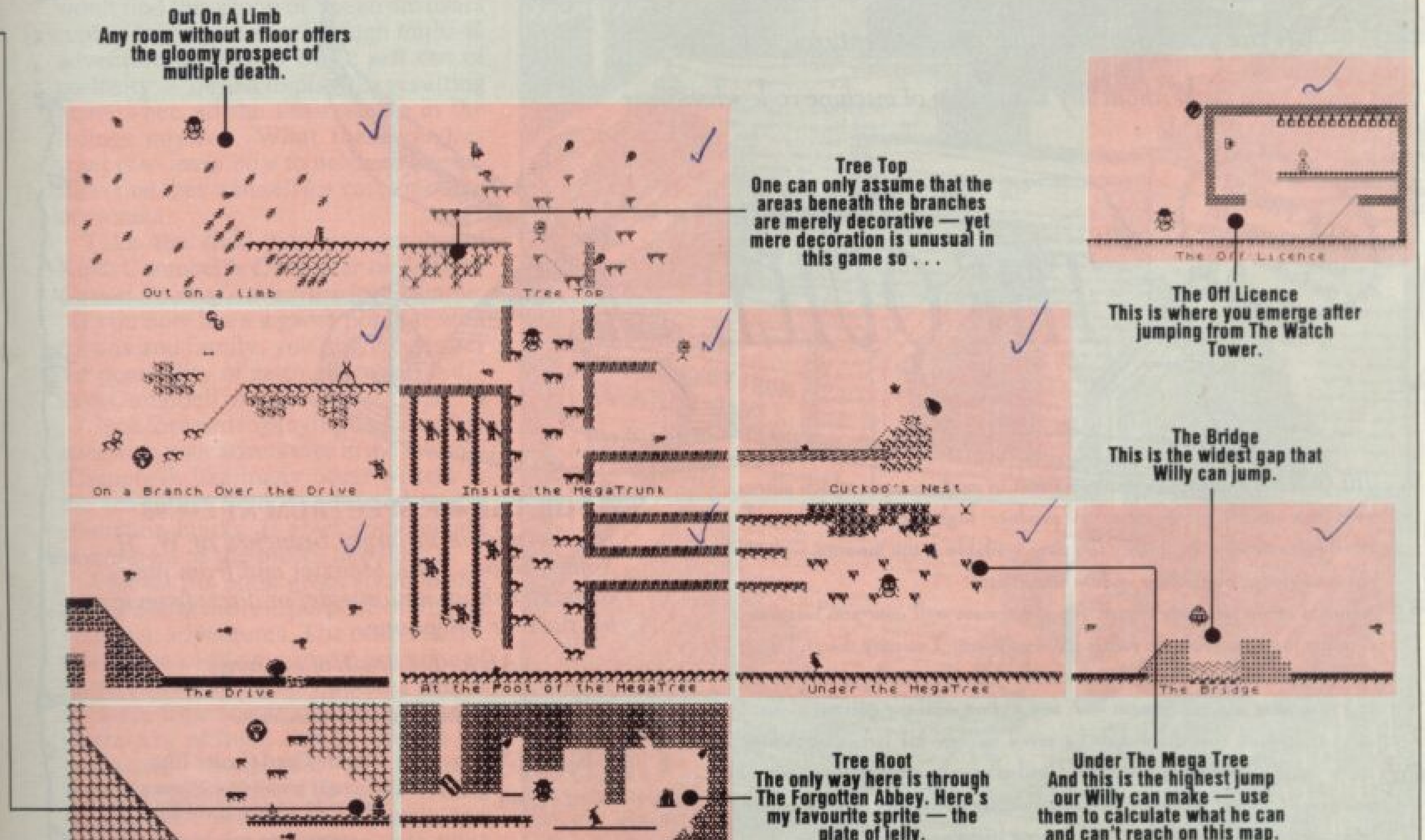
Along with the program and all its data are some very weird bytes indeed. For instance, the data for a further screen which contains very little — and, in fact, it's possible to reach it without cheating — is called ']' and just appears to have been forgotten about. For reference, it lies above the Conservatory Roof. There are also some strange-looking instructions towards the end of the program that appear to address a very complicated piece of hardware; exactly what, I don't know.

Incredibly, the data for each screen is stored in only 256 bytes — 128 for the room's appearance, 32 for its title and the remaining bytes for sprite information.

There are also quite a few oddities in the program, such as when Willy turns into what appears to be a flying pig. In fact, using a few more POKEs, being the pig character all the time can be a distinct advantage on some of the screens (sorry, not in the Emergency Generator — eat your heart out Pink Floyd!).

There are also some strange sections in some of the rooms that are either impossible to get to, or are seemingly useless. In particular, I would question the need for the gap at the bottom right of Nomen Lumi and the useless exit on the right of the Emergency Generator. As well as all this, what is the subtle pun behind 'Dr Jones Will Never Believe This'? What is 'Nomen Lumi'? And what the devil is a 'Quirkafleeg'? — and is the act of performing one illegal below a certain age? The regular staff at *YS* and myself would also be glad of any reader's help solving the mystery of the Banyan Tree — it's obvious you've got to get on top of the right-hand side of it to approach the goodies in the Conservatory Roof, but none of us can get there without fiddling it — and apparently Matthew Smith has only done it once himself, so what chance have we mere mortals?

Jet Set Willy is a great game to play, and from a programmer's point of view the ideas are structured magnificently. Even with my custom version I've only managed to get hold of around 40 objects! My excuse is that I've spent most of my time fiddling around inside the listing, what's yours? Hack on, my friends!



Obsessed by adventure, Peter Jackson girds his loins and boldly goes where few reviewers have gone before — to the shelves of the local bookstore. His mission? To discover what goes to make a prime adventure tutor for the Speccy.

NOVEL ADVENTURES

Adven'ture n. — Remarkable happening; enterprise; risk; bold exploit; commercial speculation.

— Collins Gem English Dictionary.

To start with, here's a small competition for adventure game players: spot the bit of the above definition that rarely crops up in the dungeons. Correct. Part two: spot the bit that has brought us so many books on Spectrum adventures. Correct again.

But the commerical speculation in the adventure book world is more subtle than the usual bandwagon-jumping that's done by authors and publishers, though that is certainly going on (note Melbourne House's tome explaining *The Hobbit* 'help' messages, for one). No, these books are also an appeal to reader's commercial interests. You won't find this group of Spectrum users rubber-keying its way through multi-K adventure listings in Basic just out of curiosity — or just to play the resulting game when all the answers are in the listings anyway. What these readers want is to learn how to produce adventures that they can sell for vast amounts of money.

Take the final few paragraphs of Keith Campbell's *Computer and Video Games Book of Adventure* for example. "If you now have a game popular with friends and family, you might consider the possibility of getting it published," says Campbell. And a few lines later: "I . . . look forward to playing and possibly reviewing your adventures in the future!" (Campbell, like many other writers in the adventure field, is addicted to exclamation marks. It gives some ersatz excitement to the prose, like this!!!)

At least his book gives some useful advice about the real purpose of writing your own adventures. The others under review are a little more coy, with one of them remarking vaguely that the author is "sometimes tempted to think that the mercenary attitude shines through on occasions!" (note that '!'). Perhaps several months should be spent writing an adventure just to amuse family and friends and to impress them with your programming prowess and resistance to boredom?

It's interesting to note that Campbell and most of the other authors in this book batch have had adventures published by some games house or other. Peter Gerrard, for example, is described in the blurb of his *Exploring Adventures on the Spectrum 48K* as the "author of two top-selling adventure games for the Commodore 64" (?). And Roy Carnell, author with Tony Bridge of *Spectrum Adventures*, has his own eponymous adventure software house.

THRILLS FOR SALE

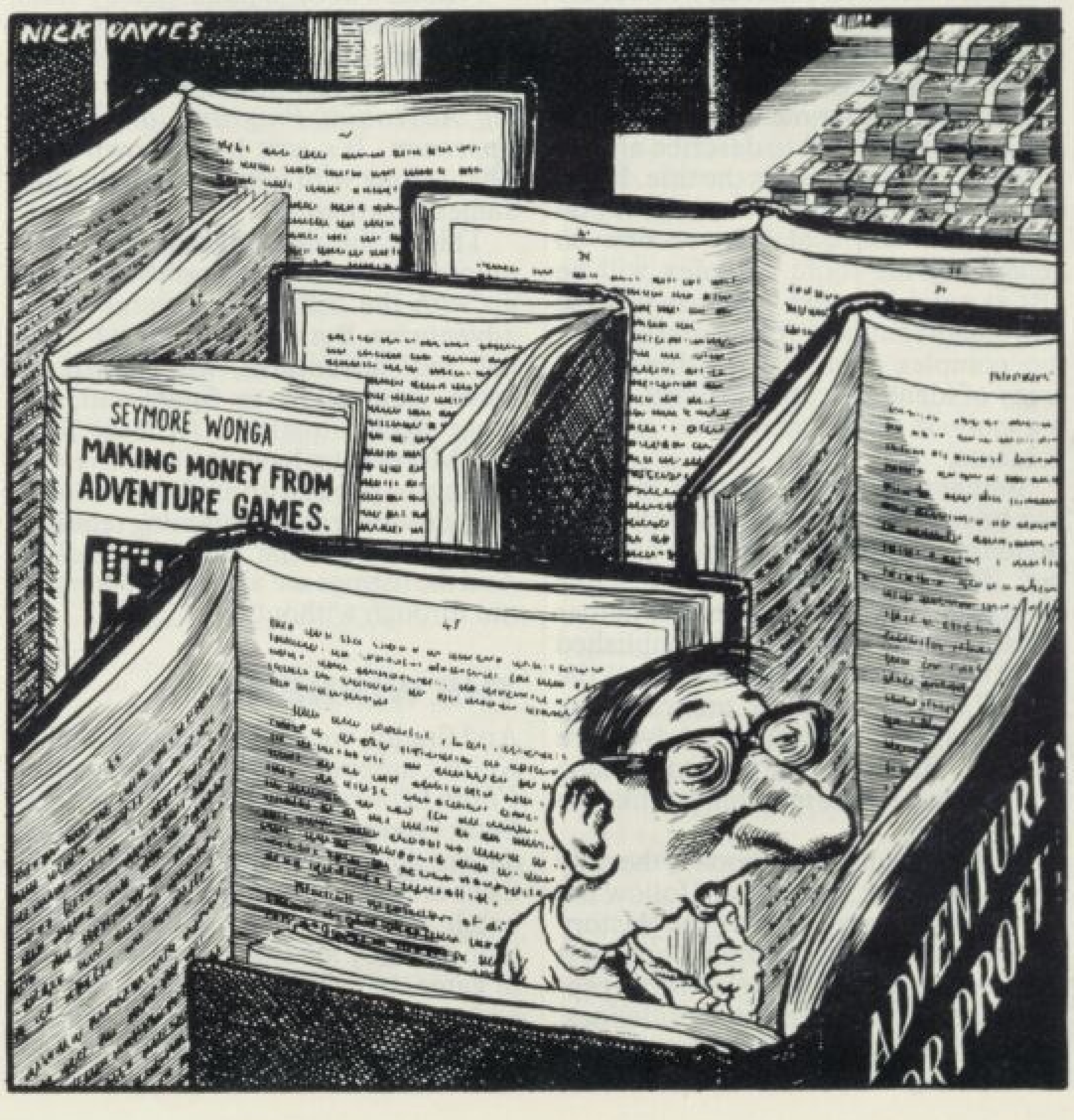
To paraphrase Dr Johnson; "Sir, no-one but a fool ever wrote an adventure except for money." So — fair enough — I decided to examine the five books from the point of view of making money; to be honest, I'm by no means immune to the temptation of writing a 'top-selling adventure' myself.

One thought struck almost immediately — the programming in adventures is not very difficult. Long, certainly; convoluted, yes; but not actually difficult. The main purpose in adventures is to understand the player's typed input and relate this to stored information about objects, locations and problems in the fantasy world created by the programmer. (At least, that's what *I* think it is. We'll be coming back to what some of the *authors* think it is later.) Programming this requires a bit of string slicing, a lot of checking against arrays, some setting and testing of flags, and a whole raft of IF-THEN-GOTOs and ON-GOTOs. The main problem is continuity, the linking together of locations, objects and actions, but that's a debugging and play-testing problem rather than a programming one.

What stops people writing adventure programs, then, is not so much difficulty as problems of scale. Adventures are so darn *big* that they can daunt the beginner; warnings in the books about what you can do if you run out of memory are more frightening when you have 48K than when you only have 16K. All the authors have addressed themselves to this confidence problem in various ways — and with varying degrees of success.

THE SHAPE'S THE SAME

Despite differing approaches to the actual programming, all the books seem to follow a set pattern. First there is an explanation of fantasy and role-playing games and that's followed by a description of *Dungeons and Dragons* with all the obligatory copyright and trademark blurb from TSR. Then comes an outline study of the original mainframe adventure, descriptions of the early micro ver-



sions, and eventually the author gets into his own programs.

About the weirdest candidate for this stock structure comes in *Spectrum Adventures* by Bridge and Carnell. The usual stuff about Crowther and Woods, Scott Adams *et al*, is just as expected, and there are descriptions and examples of such classic text adventures as *Colossal Cave* and Level 9's *Dungeon Adventure*. There's even a separate chapter on *The Hobbit*. Then the authors start programming and the game they have chosen to illustrate adventures is a pure graphic, 'move about on the screen and belt monsters', arcade-style thing that bears little relation to the heritage they are claiming. It's called *Eye of the Star Warrior*, and those fearing the mammoth task of typing it all in will be pleased to learn that it's also available on cassette.

Chapter six of *Spectrum Adventures* offers some self-justification for using a 'graphic adventure' (the phrase is theirs, I'd prefer 'arcade knockabout') to demonstrate adventure programming technique. "In this way we can present the maximum number of techniques", it says. "The second reason is simple — a text adventure, while being a lot of fun to play, is not much of a surprise after being typed in from a listing!", it goes on. Leaving the exclamation mark aside, this seems to me like turning a bug into a feature. The most famous adventure programs ever are all pure text (leaving aside *The Hobbit* which is a very mediocre adventure indeed without the pictures) and giving the thrill-thirsting public something that isn't really an adventure at all is a bit much.

Pardon my continued ranting, but if this book was meant to describe arcade games it should say so in the title. It will not do for the authors to say, as they do in Chapter six, that "the section on generating the room complex will be just as valid in a text game, as will the movement routines", particularly when the 'complex' consists of a three-level array of identical cubicle rooms differing only in their monstrous graphic contents.

Similar criticisms, only less spiteful, can be made of a book I haven't mentioned before: Andrew Nelson's *Creating Adventure Programs on Your Computer*. We're not told whether Nelson has his trotters in the published game trough, only that he has spent "the last 18 months devising, programming and playing computer Adventure games". But it seems safe to assume that he's in it for the moolah, just like the rest of us.

Be that as it may, his book is the only one in the stack that doesn't follow the standard pattern. There's no history lesson, but before jumping headfirst into the programming, Nelson does provide one or two interesting sidelines. First he gives a little non-computer adventure game, reminiscent of Penguin's *Fighter Fantasy Gamebooks* in its programmed learning course layout. This forms the basis of his first game listing, *Werewolves and Wanderer*, and he also shows a printout of the program actually running — a feature other authors should copy.

Nelson gives three games listings, all based on the same techniques and increasing in complexity through the book. Only the first has its programming described in detail, with the code split into short sections and explained. It's left as an exercise to the reader to figure out how these techniques are used in the later games, but there's nothing too difficult. All three have much more in common with the adventure classics than Bridge and Carnell's example.

But that's not to say that Nelson's work is entirely kosher. As the sample runs show, the aims of the games are to explore an environment, pick up treasure and kill monsters. True, the room descriptions will be more familiar to the old-timers, but the player's vocabulary is limited to a set of one-letter commands and the puzzle-solving aspect of adventures is sadly neglected in favour of 'slash and bash'.

All three games were created on the IBM PC, but since they are text-only there are few problems in converting them to run on the Spectrum (or anything else for that matter), and the largest of them only sucks up around 18K.

Nelson's book is published by Interface — which made it quite a surprise to find another Interface book in the stack, and one sporting a similar title: *Creating Adventure Programs on the ZX Spectrum*. It's writ by our own Peter Shaw along with fellow infant prodigy, James Mortleman.

This is the most lightweight of the quintet, the bulk of the book being taken up with the listings of several simple adventures. But these are what I would call *real* adventures, with puzzles to solve, equipment to figure out, and other stock situations. At least Shaw and Mortleman have resisted overburdening the pages with explanation of the straightforward programming required; the listings tell the story themselves. And the listings are also short enough to run through without fear.

THE WINNERS

And finally, we come whizzing back to Keith Campbell and his *Computer and Video Games Book of Adventure*, and Peter Gerrard's *Exploring Adventures on the Spectrum 48K*. To my mind, these are easily the two best books in the batch. Both written by published game authors, they are traditional in the type of game they describe and they clearly and succintly explain each stage in the programming of a traditional adventure.

Both follow stock format, with history and brief descriptions of published games at the front, followed by programming details for one game on three machines (Campbell), or three games on one machine (Gerrard).

But Gerrard's book wins it, and not just because he has the advantage of being able to concentrate on the Spectrum alone. Demo or no demo, Campbell's adventure program is trivial, and if you produced a similar game using the techniques given you'd be laughed out of every software house waiting room in the land. Gerrard's, on the other hand, are the real thing. They're of the right commercial scale and — once again — it's no surprise to find the games available on cassette from the book publisher; with changes in locations and objects you could well have a sellable game on your hands. Gerrard also gets high marks for providing a set of scenarios that the reader *could* use to get a foot in the door of those mirth-ridden software houses.

One final point (and really an admission of prejudice) both Campbell and Gerrard produce 'real' adventures and their books have also proved a few points to my satisfaction. First, lots of the adventure games you come across are nothing of the sort. Second, writing adventure programs is a lot more difficult to start and finish; the actual coding seems a piece of cake. And third, I now know that I could write a wonderful adventure game if I only had the time. . . or perhaps a good adventure book?

One last gripe from someone who is within 'the biz'. Any publisher who allows the sort of production cock-ups that have marred all the five books here — typographical errors, mispellings, misregistered printing, misbound and repeated sections of pages, and mis-just-about-everything-else — would not keep my valued custom. And yet all these books cost more than a reasonably good adventure cassette. 'Commercial speculation' is right, and so, nearly, was Dr Johnson. YS

WE LOOKED AT. . .

Computer & Video Games Book of Adventure by Keith Campbell, Melbourne House.
ISBN 0-86161-143-8, £5.95.

Exploring Adventures on the Spectrum 48K by Pete Gerrard, Duckworth.
ISBN 0-7156-1796-6, £6.95.

Spectrum Adventures by Tony Bridge & Roy Carnell, Sunshine.
ISBN 0-946408-07-6, £5.95.

Creating Adventure Programs On Your Computer by Andrew Nelson, Interface.
ISBN 0-907563-36-8, £4.95.

Creating Adventure Programs on the ZX Spectrum by Peter Shaw & james Mortleman, Interface.
ISBN 0-907563-58-9, £4.95.

ADDING ZIP 2! CONTINUED FROM PAGE 50

```
netab-2: GO SUB deek
6760 IF n=t THEN  LET t=t1*4+linetab-2:
RETURN
6765 IF n>t THEN  LET u=t1-1: GO TO 6750
6770 LET L=t1+1: GO TO 6750
6790 REM
6795 REM Correct line references (patch)
6800 FOR q=linetab-2 TO tp STEP -2: LET
t=q: GO SUB deek: LET i=t: GO SUB deek:
LET n=t: GO SUB find line: LET t=t-2: GO
 SUB deek: GO SUB doke: NEXT q: RETURN
6895 REM Put address of line t at pc-1 (
request)
6910 IF lnum<t THEN  GO TO 6920
6915 LET n=t: GO SUB find line: LET t=t-
2: GO SUB deek: LET i=pc-1: GO SUB doke:
 RETURN
6920 IF tp-2<=pc THEN  GO TO too big
6925 LET tp=tp-2: LET i=pc-1: GO SUB dok
e: LET i=tp: LET t=pc-1: GO SUB doke: RE
TURN
6990 REM
6995 REM **** SYNTAX PARSER (pass2)
```

Lines 7000-7695 Syntax Parser. This converts the Basic symbols (other than calculations) into intermediate codes.

```
7000 GO SUB init: LET bugs=false: LET pe
ep=0: LET xref=top-2: LET z=0
7010 LET pc=pc-1: LET c=68: GO SUB Z80:
REM Trap BREAK
7015 LET t=bottom+134: GO SUB deek: LET
t1=t: LET i=t1+1: LET t=vars: GO SUB dok
e: LET i=t1+9: LET t=prog-vars-1: GO SUB
 doke: REM Adjust library CLEAR code to
initialise arrays (if any)
7020 LET c=49: GO SUB Z80: LET c=10: GO
SUB Z80: REM CLEAR & CLS
7025 CLS : PRINT  PAPER 1;"COMPILING LIN
E      (0    bytes)"
7030 REM nextln
7040 IF lnum>=eof THEN  LET lnum=9999: G
O SUB addln: LET c=53: GO SUB Z80: POKE
pc,255: GO TO 5840: REM End with 'OK' me
ssage
7045 PRINT AT 0,15; FLASH 1;lnum: GO SUB
 addln
7050 REM nextst
7080 GO SUB gets
7100 IF s<>rem THEN  GO TO 7120
7105 LET s=PEEK p: LET p=p+1: IF s<>ente
r THEN  GO TO 7105
7110 LET p=p-1: GO TO pre colon
7120 IF s<>goto AND s<>gosub THEN  GO TO
 7150
7125 LET c=4+(s=gosub): GO SUB gets: IF
s=number THEN  LET t=num: GO SUB Z80: GO
 TO pre colon
7130 LET bugs=number: GO TO p2error
7150 IF s=let THEN  GO SUB gets: GO SUB
store: GO SUB gets: GO SUB maths: LET c=
assmod: LET t=assvar: GO SUB Z80: GO TO
atcolon
7175 IF s=return THEN  LET c=6: GO SUB Z
80: GO TO pre colon
7185 IF s=if THEN  GO SUB gets: GO SUB m
aths: LET c=48: LET t=lnum+1: GO SUB Z80
: GO TO nextst
7205 IF s=stop THEN  LET c=53: GO SUB Z8
0: GO TO pre colon
7210 IF s<>for THEN  GO TO 7245
7215 GO SUB gets: LET assvar=s: GO SUB g
ets: GO SUB gets: GO SUB maths: GO SUB g
ets: GO SUB maths: IF s=step THEN  GO SU
B gets: GO SUB maths: GO TO 7230
7225 LET c=2: LET t=1: GO SUB Z80: REM G
enerate implicit STEP 1
7230 LET c=7: LET t=assvar-65: GO SUB Z8
0: GO TO atcolon
7245 IF s=next THEN  GO SUB gets: LET c=
8: LET t=s-65: GO SUB Z80: GO TO pre col
on
7260 IF s<>border AND s<>pause AND (s<in
k OR s>over) THEN  GO TO 7295
7265 LET assmod=s-197: IF s=border THEN
 LET assmod=19
7266 IF s=pause THEN  LET assmod=58
7270 GO SUB gets: GO SUB maths: LET c=as
smod: GO SUB Z80: GO TO atcolon
7295 IF s=poke OR s=out THEN  LET assmod
=12+2*(s=out): GO SUB gets: GO SUB maths
: GO SUB gets: GO SUB maths: LET c=assmo
d: GO SUB Z80: GO TO atcolon
7315 IF s=clear OR s=cls THEN  LET c=10+
39*(s=clear): GO SUB Z80: GO TO pre colo
n
7320 IF s<>print THEN  GO TO 7500
7323 LET sep=false: LET c=52: GO SUB Z80
7325 GO SUB gets
7327 REM Check print item (itm)
7330 IF s=enter OR s=colon THEN  GO TO 7
480
7334 LET sep=true
7335 IF s=semi THEN  GO TO 7325
7340 IF s=appos THEN  LET c=2: LET t=13:
 GO SUB Z80: LET c=47: GO SUB Z80: GO TO
 7325
7345 IF s=comma THEN  LET c=2: LET t=6:
GO SUB Z80: LET c=47: GO SUB Z80: GO TO
7325
7346 LET sep=false
7355 IF s=chr THEN  GO SUB maths: GO TO
itm
7365 IF s<>quote THEN  GO TO 7385
7370 LET sptr=num
7375 GO SUB parsestr: IF num THEN  LET t
=sptr: LET c=26: GO SUB Z80
7376 IF more THEN  LET t=quote: LET c=2:
 GO SUB Z80: LET c=47: GO SUB Z80: LET s
ptr=nptr+1: GO TO 7375
7377 LET p=nptr: GO SUB gets: GO TO itm
7385 IF s=at THEN  GO SUB gets: GO SUB m
aths: GO SUB gets: GO SUB maths: LET c=3
4: GO SUB Z80: GO TO itm
7405 IF s=tab THEN  GO SUB gets: GO SUB
maths: LET c=35: GO SUB Z80: LET c=2: LE
T t=0: GO SUB Z80: LET c=47: GO SUB Z80:
 GO TO itm: REM Oddly, TAB requires a 2
byte parameter!
7425 IF s>=ink AND s<=over THEN  LET ass
mod=s: GO SUB gets: GO SUB maths: LET c=
assmod-189: GO SUB Z80: GO TO itm
7440 GO SUB maths: LET c=27: GO SUB Z80:
 GO TO itm
7480 IF NOT sep THEN  LET c=2: LET t=13:
 GO SUB Z80: LET c=47: GO SUB Z80
7485 GO TO atcolon
7500 IF s<>input THEN  GO TO 7535
7505 GO SUB gets
7507 IF s=semi OR s=comma THEN  GO TO 75
05
7515 IF s=colon OR s=enter THEN  GO TO a
tcolon
7520 IF s>=capital A AND s<=capital Z TH
EN  GO SUB store: LET c=9: GO SUB Z80: L
ET c=assmod: LET t=assvar: GO SUB Z80: G
O TO 7507
7525 LET bugs=input: GO TO p2error
7535 IF s=random THEN  GO SUB gets: GO S
UB maths: LET c=57: GO SUB Z80: GO TO at
colon
7635 GO SUB skip st: GO TO atcolon
7660 REM pre colon
7665 GO SUB gets
7670 REM atcolon
7675 IF s=colon THEN  GO TO nextst
7680 IF s=enter THEN  GO TO nextln
7685 GO TO 7130: REM Computed  GO TO / G
O SUB error??
7687 REM store (var. name in s)
7690 LET assvar=s-65: GO SUB gets: LET a
ssmod=11: IF s=lbracket THEN  GO SUB get
s: LET assmod=13: GO SUB maths: GO SUB g
ets: IF a(assvar+1)=a(assvar+2) THEN  LE
T bugs=index: GO TO p2error
7695 RETURN
7698 REM
```

Lines 7700-7990 Maths Evaluator. This converts calculations into intermediate codes.

```
7700 REM **** EXPRESSION HANDLER
7750 REM exppush
7752 IF z=maxz THEN  GO TO p2error
7755 LET z=z+1: LET S(z)=tsav: LET L(z)=
s2: LET oplast=true
7760 RETURN
7765 REM exppop
7770 LET z=z-1: RETURN
7780 REM Scan expression (maths)
7785 LET commas=0: LET z=0
7790 LET s2=opbrt: LET tsav=0: GO SUB ex
ppush: REM mark start
7795 REM expnext
7800 LET s2=s: IF s2<>number THEN  GO TO
 7805
7802 IF L(z)=uminus THEN  LET num=-num:
GO SUB exppop: GO TO 7802: REM Perform n
egation at once
7803 LET c=2: LET oplast=false: LET t=nu
m: GO SUB Z80: GO TO expnexts: REM Put n
um on Z80 stack
7805 IF s2>capital Z OR s2<capital A THE
N  GO TO expop
7810 LET oplast=false: LET tsav=s2-65: G
O SUB gets: LET s2=s: IF s2=lbracket THE
N  LET s2=index: GO SUB exppush: LET s2=
opbrt: GO SUB exppush: GO TO expnexts: R
EM Subscript is a new expression
7815 LET c=1: LET t=tsav: GO SUB Z80: GO
 TO expnext
7820 REM expop
7822 IF s2=plus AND oplast THEN  GO TO e
xpnexts: REM Unary plus
7825 IF s2=appos OR (s2=comma AND commas
=0) OR s2=step OR s2=to OR s=then OR s2=
colon OR s2=semi OR s2=enter THEN  LET s
2=rbracket: REM End expression with ')'
7828 REM Identify operators: ()*+,-./<=>
 etc.
7830 IF s2=45 AND oplast THEN  LET s2=um
inus: GO TO expdoop
7835 IF s2<=47 AND s2>=40 THEN  LET s2=s
2-39: GO TO expdoop
7840 IF s2<=62 AND s2>=60 THEN  LET s2=s
2-50: GO TO expdoop
7845 IF s2<=201 AND s2>=188 THEN  LET s2
=s2-175: GO TO expdoop
7847 IF s2=attr THEN  LET s2=27: LET com
mas=commas+1: GO TO expdoop
7850 GO TO p2error: REM Unrecognised ope
rator in s2
7855 REM expdoop
7857 IF z=1 AND s2=clbrt THEN  RETURN :
REM finished
7860 LET prio=P(L(z)): IF L(z)=opbrt THE
N  LET prio=0
7865 IF oplast OR prio<P(s2) OR prio=P(s
2) AND oplast THEN  GO SUB exppush: GO T
O expnexts: REM Don't operate yet
7875 IF L(z)=opbrt AND s2=clbrt THEN  GO
 SUB exppop: GO TO expnexts: REM End of
sub-expression
7880 IF L(z)=opbrt THEN  GO TO p2error
7885 LET c=O(L(z)): LET t=S(z): GO SUB e
xppop: IF c<>-1 THEN  GO SUB Z80: GO TO
expdoop
7887 LET commas=commas-1: GO TO expdoop
7890 REM expnexts
7900 GO SUB gets: GO TO expnext
7910 REM parsestr
7915 LET nptr=sptr: LET num=0
7920 IF PEEK (nptr)<>quote THEN  LET npt
r=nptr+1: LET num=num+1: GO TO 7920
7925 LET nptr=nptr+1: LET more=PEEK (npt
r)=quote: RETURN
7950 REM PASS 2 Error (p2error)
7955 LET errors=errors+1: LET t=xref+4:
GO SUB deek: PRINT AT errors+1,0;t;" ";:
 REM Echo the most recent line reference
7956 IF bugs=index THEN  PRINT "ARRAY ";
CHR$ (assvar+65);"() WAS NOT DIMENSIONED
": GO TO 7963
7957 IF bugs=input THEN  PRINT "WRONG IN
PUT FORMAT": GO TO 7963
7958 PRINT "CALCULATION ";: IF z=maxz TH
EN  PRINT "TOO COMPLEX": LET z=1: GO TO
7963
7960 IF bugs=number THEN  PRINT "NOT ALL
OWED": GO TO 7963
7961 PRINT "NOT UNDERSTOOD"
7963 LET bugs=false
7964 IF s<>enter THEN  GO SUB fetch: GO
TO 7964
7965 IF errors<15 THEN  GO TO nextln
7967 STOP
7970 REM skip st
7975 IF s<>colon AND s<>enter  THEN  GO
SUB gets: GO TO 7975
7980 RETURN
7985 REM gets
7986 IF NOT bugs THEN  GO SUB fetch
7987 RETURN
7990 REM
7995 REM **** Z80 CODE GENERATION
```

Lines 8000-8500 Code Generation. This converts intermediate codes into Z80 machine code.

```
8000 IF errors THEN  RETURN
8010 IF pc>prog+36 THEN  PRINT AT 0,21;
PAPER 1;pc-prog-36
8020 IF c<=5 OR (c>=7 AND c<=8) OR c=11
OR c=13 OR c=26 OR c=37 OR c=48 OR c=68
THEN  GO TO 8100
8025 REM Store code routine 'c'
8030 IF c>=28 AND c<=33 THEN  LET c1=c:
LET c=35: GO SUB dump: POKE pc-3,c1-12:
RETURN : REM Handle INK, PAPER etc alike
8032 REM dump (template No. c)
8035 LET j=bottom+c*2: LET t=j: GO SUB d
eek: LET i=t: LET t=j+2: GO SUB deek: LE
T j=t-i
8045 IF pc+j>=tp THEN  GO TO too big
8050 LET peep=peep AND PEEK (i)=225: IF
peep THEN  LET pc=pc-2
8055 FOR t=1+peep TO j: POKE t+pc,PEEK (
i+t-1): NEXT t: LET pc=pc+j
8060 LET peep=PEEK pc=229: RETURN
8100 LET t1=t: GO SUB dump: LET t=t1
8105 IF c=1 THEN  LET i=pc-2: LET t=vars
+8*t: GO SUB doke: RETURN
8110 IF c=2 THEN  LET i=pc-2: GO SUB dok
e: RETURN
8115 IF c=3 THEN  LET i=pc-9: LET t=a(t+
1): GO SUB doke: RETURN
8120 IF c=4 OR c=5 OR c=48 THEN  GO SUB
request: RETURN
8125 IF c<>7 THEN  GO TO 8150
8130 LET t1=t*8+vars: LET i=pc-15: LET t
=t1+2: GO SUB doke: LET i=pc-11: LET t=t
1+4: GO SUB doke
8135 LET i=pc-7: LET t=t1: GO SUB doke:
LET i=pc-4: LET t=pc+1: GO SUB doke
8140 LET i=pc-1: LET t=t1+6: GO SUB doke
: RETURN
8150 IF c<>8 THEN  GO TO 8180
8155 LET t1=vars+t*8: LET i=pc-28: LET t
=t1: GO SUB doke: LET i=pc-24: LET t=t1+
2: GO SUB doke
8160 LET i=pc-20: LET t=t1: GO SUB doke:
 LET i=pc-16: LET t=t1+4: GO SUB doke: L
ET i=pc-2: LET t=t1+6: GO SUB doke: RETU
RN
8180 IF c=11 THEN  LET i=pc-1: LET t=var
s+8*t: GO SUB doke: RETURN
8185 IF c=13 THEN  LET i=pc-7: LET t=a(t
+1): GO SUB doke: RETURN
8190 IF c<>26 THEN  GO TO 8205
8192 LET i=0
8194 LET i=i+1: IF PEEK (t+i)<>quote THE
N  GO TO 8194
8196 IF i+pc>=tp THEN  GO TO too big
8198 LET i1=i: LET t1=t: LET i=pc-6: LET
 t=i1: GO SUB doke: LET i=pc-9: LET t=pc
+1: GO SUB doke
8200 FOR j=1 TO i1: POKE pc+j,PEEK (t1+j
-1): NEXT j: LET pc=pc+i1: RETURN
8205 IF c=37 THEN  LET i=pc-4: LET t=pc:
 GO SUB doke: RETURN
8210 IF c=68 THEN  LET t=bottom+c*2-2: G
O SUB deek: LET i=t+1: LET t=top-8: GO S
UB doke: RETURN
8500 PRINT  INK 0; PAPER 6;"CODE GENERAT
ION ERROR ";C: STOP
```

3D PLOTTING

PART TWO

Continuing on his quest to make perspective a reality on the Spectrum.* Damir Skrgatic *tackles the subject of symmetrical figures generated through the application of spherical co-ordinates.

In Part One (published in last month's *Your Spectrum*), the perspective was demonstrated using the 'plane plotting' routine. This time, the spherical co-ordinates are chosen to generate a variety of symmetrical 3D objects and also to provide relatively simple transformations (scaling, rotation and translation). The routines are used in the main part of the program (listed in Part One) and the co-ordinates system is depicted in Figure 2.

Object generation starts at line 90 with the set of INPUT statements defining object parameters chosen by the observer. The parameters asked for in the program will produce data for the generation of:

— the number of vertical sections, sb
— the number of sides in each section, sa
— the starting longitude; that is, the longitude of the first vertical section, b0
— the starting latitude; that is, the latitude of the origin of the DRAW vector representing the first side of the polygon, a0

This can be seen in Figure 3.

sb = 3
sa = 4
b0 = 0°
a0 = 45°

Figure 3 will generate three vertical sections — at 0°, 120° and 240° — each section being represented by a wire-frame figure of a square orientated so that it has two horizontal and two vertical sides. If you were to make a0 equal to 0°, you would produce squares with sides at 45° to the horizontal (see Figure 4).

The object size will obviously be determined by R (line 100) and the apparent size using perspective will also depend on the object distance from the observer, z0 (line 90). Varying the value of z0 is equivalent to 'zooming' with a camera — translation along depth. The x and y translation is achieved by varying x0 and y0 respectively (the centre of the screen is x0 = 0 and y0 = 0).

So now, with reference to the *3D Plotting* program published in the first part of this article (see last issue), here follows a detailed breakdown of the lines which deal specifically with spherical co-ordinates.

Lines 170 and 180 convert chosen numbers, sb and sa, into angular increments in degrees, db and da, used by the FOR-NEXT loops in lines 190 and 200.

Lines 210 to 270 convert spherical co-ordinates R, a and b into rectangular co-ordinates x, y and z (and their increments: dx, dy and dz), so that the object

Figure 3

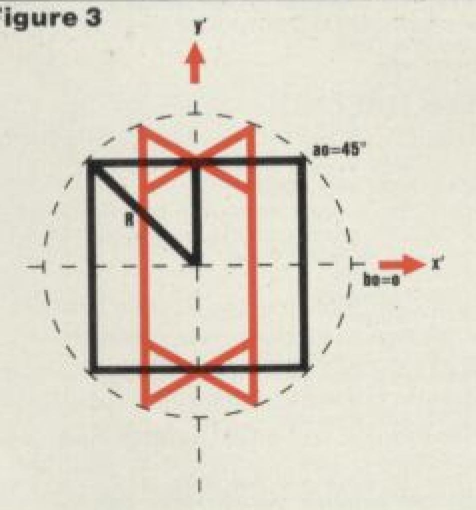


Figure 4

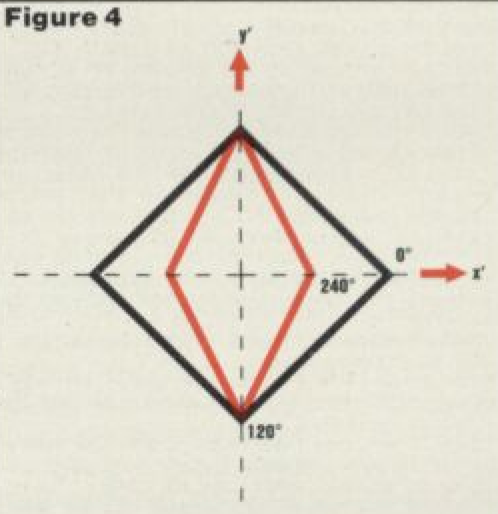
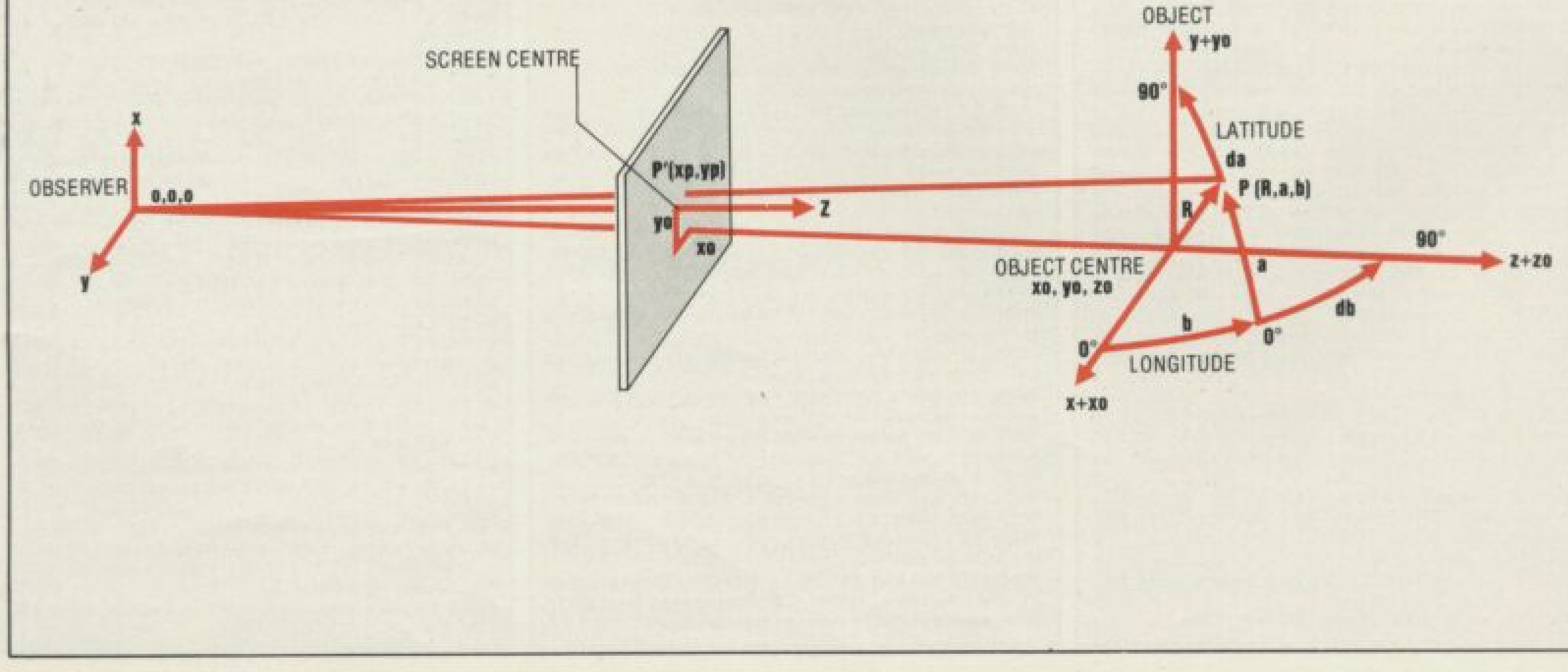


Figure 2



3D PLOTTING

PART TWO

can be plotted in perspective in a similar manner to the plane drawing routines demonstrated in Part One.

Lines 220 to 240 are standard equations converting spherical co-ordinates into rectangular co-ordinates and can be derived from Figure 2 using trigonometry. Note that all angles under SIN and COS have been converted to radians = PI/180 * angle in degrees.)

Lines 250 to 270 compute cartesian increments (vectors) dx, dy and dz, corresponding to the latitude increment, da. (This is a good example of a very cumbersome number crunching exercise which a home computer can do easily, although not as fast as CAD dedicated machines.) Note that all results are still in millimetres.

Lines 280 to 360 provide clipping outside the viewing pyramid and the limits used correspond to d = 500mm and the Spectrum screen window size as measured on a 12-inch TV (see Part One).

Line 370 is the standard perspective transformation (Equation 1) used in the subsequent PLOT statement in line 380. Note that millimetres are converted into pixels in line 380.

The expression at line 383 has already been mentioned in the 'floor' subroutine in Part One; dk represents the decrement of scale factor (d/z) when the depth increases by dz. Line 390 is rather more complicated than we have met so far and as such is fully explained in the Appendix under 'vector' projections.

It'll suffice here just to explain the combined role of two different terms which make up dxp:

— dx * d/z is the perspective component due to an increase in the vector, dx

— x * dk is the perspective component due to the vector, dz

— dx * d/z is the perspective component due to an increase in the vector, dx

— x * dk is the perspective component due to the vector, dz

The role of dyp can be explained in a similar fashion.

Line 410 draws the perspective projection of a combined vector due to dx, dy and dz, in pixels. Lines 420 and 430 complete the FOR-NEXT loops for latitude and longitude stepping. Line 440 jumps to the beginning of the main program, asking for parameters of the next object to be drawn.

A few examples of modelling with spherical co-ordinates to form 'objects' were given in the first part of this article. Following the explanation of how objects are generated, here is a more comprehensive list of parameters — and their effects. For reference, see Figures 1 and 2, and lines 100 to 160 of the *3D Plotting* program featured last issue:

1. Select the figure type using the variable, sa
   - sa=1 —this draws a point in space
   - sa=2 —this draws a line in space
   - sa=3 —this draws a triangle in space
   - sa=4 —this draws a square in space
   - sa=5 —this draws a pentagon in space

When you substitute a value of more than 15 for sa, you find that a circle is drawn in space.

2. Determine the figure orientation with the starting latitude, a0, in degrees.

a0 = 0 — the first (and last) figure apex on the equator
a0 = 45 — the first (and last) figure apex at latitude 45°

3. Select the number of vertical figures (sections), sb, starting at longitude, b0, where:

b0 = 0° — draws the first (and last) figure in the plane z = z0 —that is, 'head-on' parallel to the screen.

FURTHER PARAMETER EXAMPLES FOR THE 3D PLOTTING PROGRAM*

	With translation along the x-axis		With translation along the z-axis ('zoom')		
z0	800	700			
R	60	60	60	30	20
x0			50	−50	−50
y0	0	0	0	−31	31
sb	1	1	2	1	1
sa	4	4	3	4	8
b0	90	45	0	0	90
a0	45	45	0	45	0

***These parameter examples are to be used in conjunction with the 3D Plotting program published in the last issue of *Your Spectrum*.**

3D PLOTTING

A modified version of last month's listing. If you want to update, look at lines 70-80, 430 and 660-1000.

```
  10 REM Wire frame objects in
        perspective. D.SKRGATIC
  20 GO SUB 440: STOP
  30 PRINT AT 21,13;"y=-62mm";AT 11,26;"
x=90mm";AT 0,13;"y=62mm";AT 11,0;"x=-90m
m"
  40 LET p=1.41
  50 LET d=500
  60 GO SUB 510
  70 INPUT AT 0,0;"Plane intersection:v
or h";a$
  80 IF a$="h" THEN  GO TO 660
  90 INPUT "Eye-object dist. in mm,z0=";
z0
 100 INPUT "Object 'radius' in mm,R=";R
 110 INPUT "Obj. cent. offset in mm,x0="
;x0
 120 INPUT "y0=";y0
 130 INPUT "Numb. of vert. sections,sb="
;sb
 140 INPUT "Num. of sides,each sect.,sa=
";sa
 150 INPUT "Start. longitude in deg.,b0=
";b0
 160 INPUT "Start. latitude in deg.,a0="
;a0
 170 LET db=360/sb
 180 LET da=360/sa
 190 FOR b=b0 TO 360+b0 STEP db
 200 FOR a=a0 TO 360+a0 STEP da
 210 LET x=R*COS (PI/180*a)*COS (PI/180*
b)+x0
 220 LET y=R*SIN (PI/180*a)+y0
 230 LET z=R*COS (PI/180*a)*SIN (PI/180*
b)+z0
 240 LET dx=R*COS (PI/180*b)*(COS (PI/18
0*(a+da))-COS (PI/180*a))
 250 LET dy=R*(SIN (PI/180*(a+da))-SIN (
PI/180*a))
 260 LET dz=R*SIN (PI/180*b)*(COS (PI/18
0*(a+da))-COS (PI/180*a))
 270 IF x>z/5.6 THEN  LET x=z/5.6
 280 IF (x+dx)>z/5.6 THEN  LET dx=z/5.6-
x
 290 IF y>z/8.3 THEN  LET y=z/8.3
 300 IF (y+dy)>z/8.3 THEN  LET dy=z/8.3-
y
 310 IF x<-z/5.6 THEN  LET x=-z/5.6
 320 IF (x+dx)<-z/5.6 THEN  LET dx=-z/5.
6-x
 330 IF y<-z/9 THEN  LET y=-z/9
 340 IF (y+dy)<-z/9 THEN  LET dy=-z/9-y
 350 LET xp=x*d/z: LET yp=y*d/z
 360 PLOT xp*p+128,yp*p+87
 370 LET dk=d/(z+dz)-d/z
 380 LET dxp=dx*d/z+x*dk: LET dyp=dy*d/z
+y*dk
 390 LET e=ABS ((z-d)/200)
 400 DRAW  INK e;dxp*p,dyp*p
 410 NEXT a
 420 NEXT b
 430 GO TO 70
 440 PLOT 0,0
 450 DRAW 255,0
 460 DRAW 0,175
 470 DRAW -255,0
 480 DRAW 0,-175
 490 BORDER 5
 500 RETURN
 510 FOR z=d TO d+600 STEP 100
 520 FOR x=-90 TO 90 STEP 22.5
 530 LET xp=x*d/z
 540 LET yp=-62*d/z
 550 LET f=INT ((z-d)/200)
 560 PLOT xp*p+128,yp*p+87
 570 LET dk=d/(z+70)-d/z
 580 DRAW  INK f;x*dk*p,-62*dk*p
 590 PLOT xp*p+128,yp*p+87
 600 LET dxp=20*d/z
 610 IF x+22.5>90 THEN  LET dxp=0
 620 DRAW  INK f;dxp*p,0
 630 NEXT x
 640 NEXT z
 650 RETURN
 660 INPUT "Eye-object dist. in mm,z0=";
z0
 670 INPUT "Object radius in mm,R=";R
 680 INPUT "Obj. cent. offset in mm,x0="
;x0
 690 INPUT "y0=";y0
 700 INPUT "Numb. of  sections,sa=";sa
 710 INPUT "Num. of sides,each sect.,sb=
";sb
 720 INPUT "Start. longitude in deg.,b0=
";b0
 730 INPUT "Start. latitude in deg.,a0="
;a0
 740 LET db=360/sb
 750 LET da=360/sa
 760 FOR a=a0 TO 360+a0 STEP da
 770 FOR b=b0 TO 360+b0 STEP db
 780 LET x=R*SIN (PI/180*b)+x0
 790 LET y=R*COS (PI/180*b)*SIN (PI/180*
a)+y0
 800 LET z=R*COS (PI/180*b)*COS (PI/180*
a)+z0
 810 LET dx=R*(SIN (PI/180*(b+db))-SIN (
PI/180*b))
 820 LET dy=R*SIN (PI/180*a)*(COS (PI/18
0*(b+db))-COS (PI/180*b))
 830 LET dz=R*COS (PI/180*a)*(COS (PI/18
0*(b+db))-COS (PI/180*b))
 840 IF x>z/5.6 THEN  LET x=z/5.6
 850 IF (x+dx)>z/5.6 THEN  LET dx=z/5.6-
x
 860 IF y>z/8.3 THEN  LET y=z/8.3
 870 IF (y+dy)>z/8.3 THEN  LET dy=z/8.3-
y
 880 IF x<-z/5.6 THEN  LET x=-z/5.6
 890 IF (x+dx)<-z/5.6 THEN  LET dx=-z/5.
6-x
 900 IF y<-z/9 THEN  LET y=-z/9
 910 IF (y+dy)<-z/9 THEN  LET dy=-z/9-y
 920 LET xp=x*d/z: LET yp=y*d/z
 930 PLOT xp*p+128,yp*p+87
 940 LET dk=d/(z+dz)-d/z
 950 LET dxp=dx*d/z+x*dk: LET dyp=dy*d/z
+y*dk
 960 LET e=ABS ((z-d)/200)
 970 DRAW  INK e;dxp*p,dyp*p
 980 NEXT b
 990 NEXT a
1000 GO TO 70
```

3D PLOTTING

PART TWO

b0 = 90° — places the first figure in the plane x = x0 — that is, 'edge-on' to the screen.

This enables us to draw figures from the bundle of vertical planes — the plane intersection is placed at co-ordinates x0, z0.

The program could be modified easily to draw a bundle of planes with horizontal intersection by swapping around the longitude and latitude. We include some further examples of parameters asked for in the INPUT statements (by the program given in last month's article) which can be effective when combined with translation and rotation.

Remember that all dimensions are in millimetres and will be drawn accurately in perspective for the parts of the 3D 'object' which lie within the viewing pyramid. This means that the value of R must be less than or equal to 62 for the parts of the object which lie at z0 = 500 — that is, the screen depth.

Note that selection of z0 = 500 will make the front sections of the objects lie in front of the screen and therefore, the top and bottom parts of the object will be subject to 'clipping'.

Translation (for example, along the z-axis) can be effectively demonstrated by the addition of a z0 loop in the program:

```
90   REM 'Zoom' example
. . .
185  LET z0 = 500
. . .
440  LET z0 = z0 + 100: IF z0> 1100
     THEN GO TO 90
443  GO TO 190
```

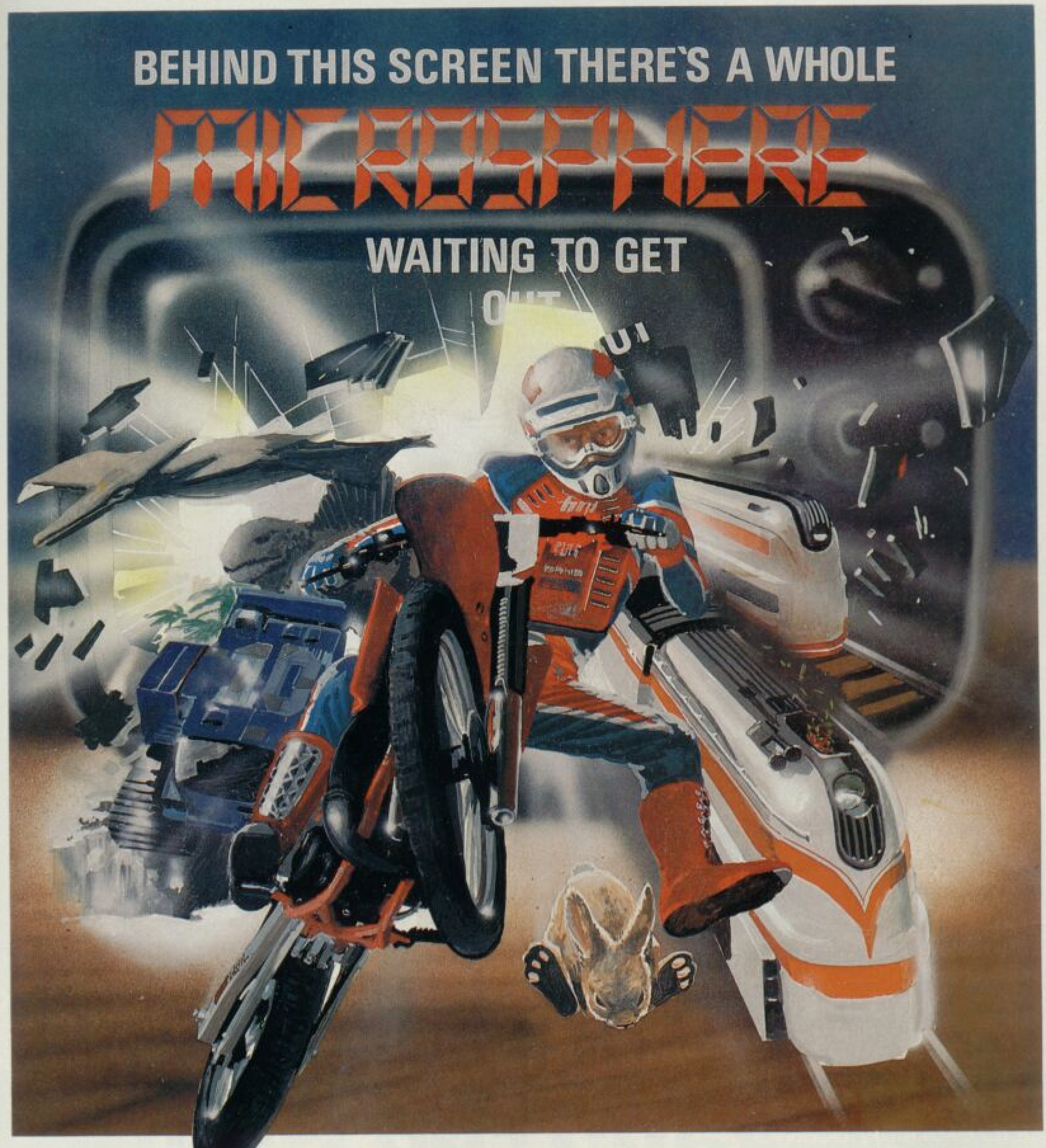
The same can be done for translation along the x and y axes.

Combined translation (in the x-axis) and rotation around the vertical axis (longitude) can be done in the following way:

```
110  REM 'Combined translation mix and
     'longitude rotation'
. . .
185  LET x0 = -45
188  LET b0 = b0 + 10: REM 'Next
     object will be rotated 10°'
. . .
440  LET x0 = x0 + 22.5: IF x0 > 45
     THEN GO TO 90
443  GO TO 188
```

	BENCH												ROUND TABLE	
	TOP				LEGS				SHELVES			CARPET/ SHADOW	TOP	BOTTOM
Z0	700	700	500	900	500	900	600	600	600	600	600	600	600	600
R	200	200	10	10	8	8	37	31	8	8	8	40	30	12
x0	-90	-70	-80	-80	-80	-80	90	80	85	85	85	0	0	0
y0	-45	-45	-45	-45	-53	-53	-40	-40	-40	-20	-50	-62	-45	-53.5
sa	1	1	1	1	1	1	1	1	1	1	1	1	1	1
sb	2	2	2	2	4	4	4	4	4	4	4	11	15	4
b0	90	90	0	0	90	90	90	90	90	90	90	0	0	45
a0	0	0	0	0	45	45	45	45	0	0	0	0	0	45
	VERTICAL (V)								HORIZONTAL (H)				(V)	

Some examples of the kind of shapes you can produce with the program given within this article. You could also try producing some of your own shapes with a twist of translation loops (R=R (y)) —have a go, they're highly recommended. YS

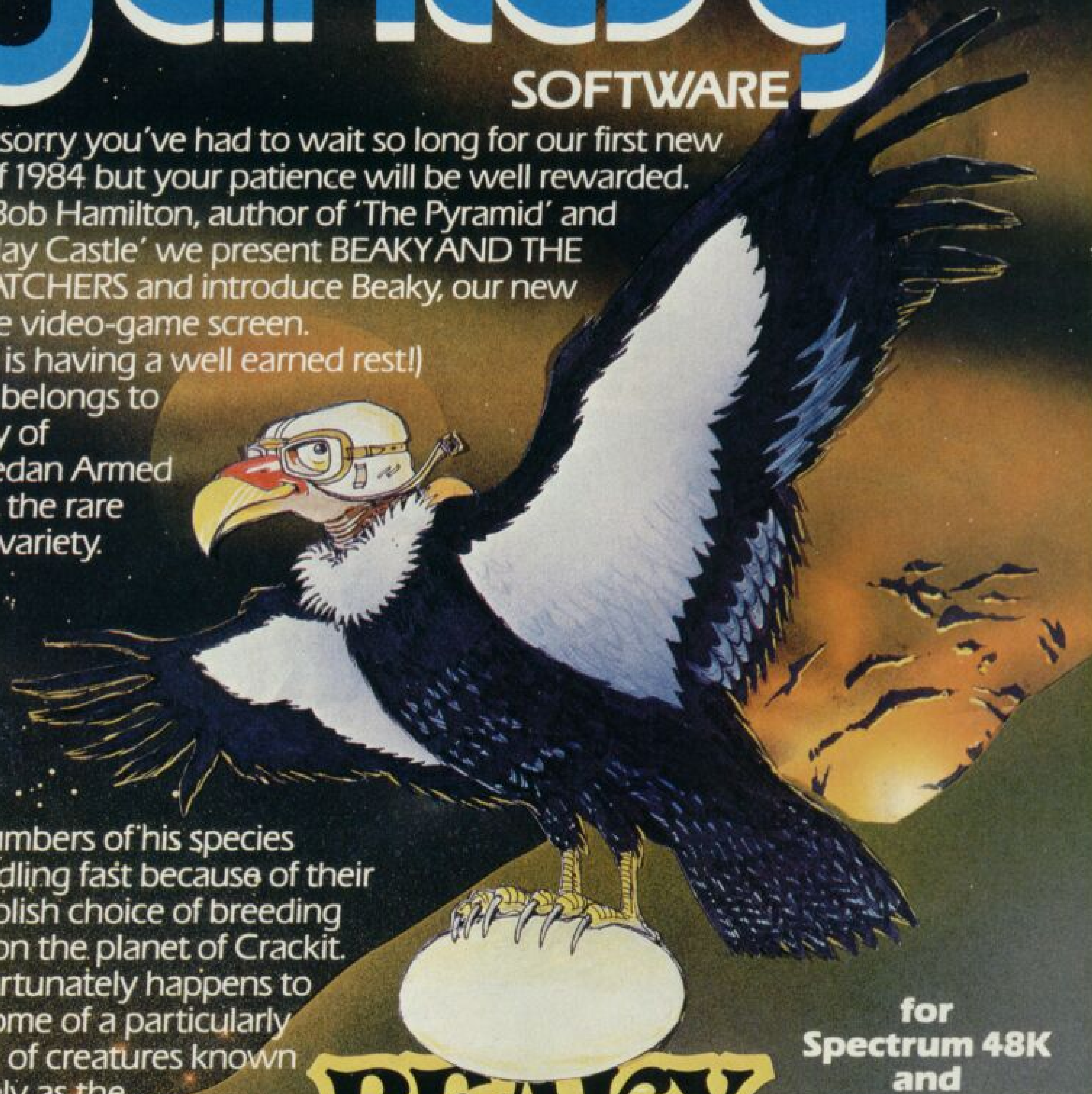

BEHIND THIS SCREEN THERE'S A WHOLE
MICROSPHERE
WAITING TO GET
OUT

fantasy SOFTWARE

We're sorry you've had to wait so long for our first new release of 1984 but your patience will be well rewarded.

From Bob Hamilton, author of 'The Pyramid' and 'Doomsday Castle' we present BEAKY AND THE EGGSNATCHERS and introduce Beaky, our new star of the video-game screen.

(Ziggy is having a well earned rest!)



Beaky belongs to the family of Andromedan Armed Condors, the rare goggled variety.

The numbers of his species are dwindling fast because of their rather foolish choice of breeding ground, on the planet of Crackit. This unfortunately happens to be the home of a particularly evil band of creatures known collectively as the Eggsnatchers. Their sole malicious intent in life is to extinctify Beaky's species by stealing or destroying the eggs by any foul means available.

BEAKY and the Egg Snatchers

for Spectrum 48K and Commodore 64

Your objective is to try and brood, hatch out and rear as many little Beakys as possible through 12 different screens of formidable but delightful arcade action.

FANTASY SOFTWARE is available from W.H.SMITHS, JOHN MENZIES, BOOTS, WOOLWORTHS, LASKYS, GREENS, RUMBELLOWS, SPECTRUM GROUP and all other good software retailers.

Beaky and the Eggsnatchers is available for 48K Spectrum and Commodore 64 at £6.50 on cassette or on disk for the Commodore 64 at £9.50 from

FANTASY SOFTWARE, FAUCONBERG LODGE, 27A ST GEORGES ROAD, CHELTENHAM, GLOS GL50 3DT
despatched by return first class post together with free membership of the Fantasy Micro Club.
Trade Enquiries welcome – telephone 0242-583661.

SPECTRUM CONTROL

Penny Page *invites you to bend your brain cells around a rather psychedelic problem.*

PROJECT 3

Drawing empty shapes with your Spectrum is a fairly simple business, but how do you set about colouring them in? The answer, of course, is a 'fill routine'. And that's exactly what this project is all about. Our fill routine has been surrounded by some additional coding just to illustrate how it works. And this draws a circle, a square and a triangle which the fill routine must then get to work on. Unfortunately, there appears to be something dreadfully wrong. Because instead of producing three solid shapes on screen, the program goes a bit haywire and comes up with what looks like a strange psychedelic pattern.

So, for all would-be space fillers, and especially those genius programmers who feel able to unravel this knotty problem, *YS* wishes you good luck. And, as ever, we will be pleased to receive your solutions, thoughts or comments which we'll be examining closely in issue 5.

```
  10 REM Filling it in
  20 LET x=100
  30 LET y=100
  40 CIRCLE 100,100,20: INK 1: GO SUB 9000
  50 LET x=150
  60 LET y=150
  70 PLOT 140,140: DRAW 20,0: DRAW 0,20:
     DRAW -20,0: DRAW 0,-20
  80 INK 2: GO SUB 9000
  90 LET x=150: LET y=50
 100 PLOT 130,52: DRAW 40,0: DRAW -20,-40:
     DRAW -20,40
 110 INK 3: GO SUB 9000: STOP
8999 REM Main subroutine
9000 LET s=255: LET st=1: LET i=0
9010 LET h=175:'LET he=INT (RND*5): LETn=0
9020 GO SUB 9500
9030 LET h=0
9040 LET he=-(INT (RND*2)+2): LET n=1
9050 GO SUB 9500
9060 LET s=0: LET st=-INT (RND*15)+5: LET
     i=-1
9070 LET h=175: LET he=1: LET n=0
9080 GO SUB 9500
9090 LET h=0
9100 LET he=-1: LET n=1
9110 GO SUB 9500
9120 RETURN
9500 FOR a=x+i TO s STEP st
9510 FOR g=y-n TO h STEP he
9520 IF POINT (a,g)=1 THEN  GO TO 9550
9530 PLOT a,g
9540 NEXT g
9550 IF POINT (a+st,y+he)=1 THEN  RETURN
9560 NEXT a
9570 RETURN
```

PROJECT 1 REVISITED

Sifting through your responses to Project 1,* Penny Page *comes up with the good, the bad... and the ugh!

Project 1, as most will recall, was the half-baked gizmo that turned the Spectrum into a sort of sketch pad —well, almost, anyway. However, not everyone seemed to understand what the thing was supposed to do — and, consequently, that led people like John Elliot and M & L Plows to come up with some unexpected — not to say — interesting solutions. Although not entirely what the doctor ordered, we'll be getting round to them at a later date — to look at the sort of thing they came up with.

And then, there were some people (Dan Hayes, Mr C Oswin and Mr R Sharman) who felt that the program was so full of 'bugs' that the only approach was to set about completely rewriting it (cheek!), but who in the process came up with some interesting-looking solutions. In many cases, this meant adding all those refinements necessary to make the program a pleasure to use; like getting the INK colour change routine to work properly, and giving a prompt for entering the starting co-ordinates. Oh, and also making the program work automatically, without the need to press a key before initialising those PLOT co-ordinates. As it was, any key other that 'Z' pressed at this stage terminated the program.

The problem, of course, boils down to the idiotic and ridiculously over-logical PLOT instruction (written this way, presumably, to catch out the unwary!). It's over-logical because it employs absolute values, which means that negative numbers are quite acceptable — until, that is, they fall below -255 and -175, respectively. Then, the PLOT instruction will detect that either one or other of the values are out of range and signal an error, and return the user to command mode. However, it's not quite that simple. For example, in Project 1, what happens when X or Y rise above 255 and 175? Again the values will be out of range and an error signalled. However, lines 1500 and 1510 prevent this from ever happening by checking to see if one or other of these figures has reached its limit and, if so, subtracting one. The program can then continue execution, albeit somewhat eratically; and this can be explained by considering the following example.

Imagine a line drawn diagonally up the screen. Both X and Y values will increase positively) as the plot progresses. However, when the top of the screen is reached, one is subtracted from the Y (vertical) co-ordinate, with the X (horizontal) co-ordinate still having some way to go before it reaches its limit. The result is a straight line drawn from left to right across the top of the screen. For when the Y co-ordinate reaches 175, one is subtracted andhis keeps the vertical position just inside the screen's limits, but X still being well below the magical 255, can continue to be incremented. This means that while Y is kept just 'on-side', X is moving towards its limit (255 represents the right-hand boundary) causing the unexpected plotting of a straight line.

Readers' Replies

By far the simplest solution came from Mr B Partridge of West Ealing, London, who managed to solve the problem by deleting line 1510 and replacing line 1500 with:

```
1500 IF X>255 OR X<0 OR Y>175
OR Y<1 THEN LET X=X—A(N):
LET Y=Y—B(N)
```

He points out that the PLOT instruction ignores the sign of its co-ordinates, and when either (or both) of these become negative, this results in those annoying 90° reflections. The control keys effectively become inverted to their intended direction, because the program is now adding A(N) and/or B(N) to X, which has become negative. So, to solve the problem, X and Y must be kept within the range of zero to their maximum values, and this is easily achieved by Mr Partridge's modification.

D Haines of Lowton, Warrington, chose to go PEEKing around, as can be seen from his amendments. And although this works perfectly well, it isn't necessarily the best approach; a listing that contains instructions such as these can often be hard to understand at some later stage:

```
1500 IF X>253 THEN LET X=X—1
: LET Y=PEEK 23678
1505 IF X<1 THEN LET X=X+1 :
LET Y=PEEK 23678
1510 IF Y>174 THEN LET Y=Y—1
: LET X=PEEK 23677
1520 IF Y<1 THEN LET Y=Y+1 :
LET X=PEEK 23677
```

But back now to the more anarchic replies and, in particular, M & L Plows of Almwch, Anglesey, who decided that it was, in fact, desirable for the program to produce a 90° reflection when the plot 'hit' the edge of the screen; they supplied the following amendments to produce this effect:

```
130 LET FH=1 : LET FV=1
1100 IF INKEY$=C$(N) THEN LET
X=X+(A(N)*FH) : LET
Y=Y+(B(N)*FV :
GOTO 1500
1500 IF ABS X>254 THEN LET
FH=FH* —1
1510 IF ABS Y>174 THEN LET
FV=FV* —1
```

Thank you, and good night! On the other hand, John Elliot of Dunnington, York, decided to stop the program from 'crashing' by preventing either of the co-ordinates from falling below zero, but allowed the erratic plotting to continue. And this, he did as follows:

```
1500 IF X>254 THEN LET X=254
1510 IF Y>175 THEN LET Y=175
1520 IF X<0 THEN LET X=0
1525 IF Y<0 THEN LET Y=0
```

He also got the starting co-ordinates prompt to appear on the screen automatically by adding line 130 GO SUB 2000. However, nobody thought to play around with the INK setting routine, which can easily be made more useful by adding the following line:

```
1350 IF INKEY$="I" THEN GO SUB
3000
```

So now, if 'I' is pressed at any time during program execution, the message 'INK=' will appear, allowing the user to change the colour of the plot, and thus permitting the drawing of multi-coloured pictures on screen.

Finally, although space prevents any nitty gritty comment, my thanks also go to BD Berman of Burton-on-Trent, RG Sharman of Ilchester, Somerset, and Dan Hayes of Poole, Dorset, for their contributions to the pot. Any further thoughts on Project 1 are still gratefully received; meanwhile, I'll be interested to find out how things are going with Project 2 — the one on the subject of key-press timing!

Ideas, amendments, project suggestions and abusive letters — all are gratefully received. Post 'em to Penny Page, (Projects), Your Spectrum, *14 Rathbone Place, London W1P 1DE.* YS

PROGRAM POWER

***If you've forgotten where a program is supposed to start then the Header Reader from* Alexander Livshit *will help you find it. Also if you're a fan of Noughts and Crosses, bend your mind with this version from* Mark Burton.**

HEADER READER

By Alexander Livshits (16/48K)

Here's a routine that should prove useful to all Speccy users — for the Basic programmer who's forgotten where a program is supposed to start, the machine codist sizing up a certain piece of code and, of course, the Microdriver attempting the tricky task of converting software for operation on cartridge.

When a program file is saved to tape it is stored in two blocks: the first part, known as the file header, is always 17 bytes long and tells the Speccy all it needs to know about the program block which follows.

This routine reads the header into high memory and decodes it into a more meaningful form which is then displayed on-screen for the user. The information provided is: the name of the program; the type of program (Basic, code block, numerical or character array); the destination address (if you are examining a code block) or the program area (for Basic); the variables area; and the autostart line number (if any).

Should the program not find a file header, the message 'NOT A HEADER' is dutifully printed on-screen.

```
   10 REM ***HEADER READER***
   20 REM
   30 REM  by A.LIVSHITS
   40 REM
   60 CLEAR 27999: REM lower RAMTOP
   65 LET s=28000: LET d=s+14
   70 DEF FN f(x)=PEEK x+256*PEEK (x+1):
 REM f(x)=double byte value at x
   80 LET dh=INT (d/256): LET dl=d-256*dh
```

Lines 60-80 The 'initial set-up' routine in which RAMTOP is lowered to 27999. The two variables 's' and 'd' set up the machine code start address and the address to which the header is to be loaded, respectively. The function defined in line 70 performs a double byte PEEK (ie. this would be a DEEK on other computers).

```
   90 DATA 55,62,0,221,33,dl,dh,17,17,0,2
05,86,5,201: REM machine code data
   95 FOR i=0 TO 13: READ m: POKE (s+i),m
 : NEXT i: REM poke in machine code
```

Lines 90-95 Read in and POKE the machine code data to the address reserved for it.

```
  100 REM -MAIN PROGRAM-
  110 CLS
  120 POKE d,4
  130 PRINT AT 10,9;"START THE TAPE"
  140 RANDOMIZE USR s: REM call machine c
ode
  150 CLS : LET type=PEEK d: IF type=4 TH
EN  GO TO 600: REM check header type
```

Lines 100-150 The 'read header' routine. This section of the program calls the machine code which reads the header into memory from location 'd' onwards. Line 150 sets the variable type to be equal to the first byte of the header.

```
  159 REM make a$="header name"
  160 LET a$=""
  170 FOR i=1 TO 10
  180 LET a$=a$+CHR$ PEEK (d+i)
  190 NEXT i
  200 PRINT "Name _____";a$: PRINT
```

Lines 159-200 Read in the header name from bytes 'd+1' to 'd+10' and print it.

```
  210 LET len=FN f(d+11): LET start=FN f(
d+13): LET pa=FN f(d+15)
  220 LET va=len-pa
  230 IF type THEN  GO SUB 400: GO TO 610
 : REM if not basic program skip next bit
```

Lines 210-230 Set up the following variables: 'len' — the file length; 'start' — the start address; 'pa' — the length of the variables area; and 'va' — the start of the variables area.

```
  239 REM basic program
  240 PRINT "Program type_____basic": PRI
NT
  250 PRINT "Program area_____";pa;" byte
s": PRINT
  260 PRINT "Variables area____";va;" byt
es": PRINT
  265 IF start>10000 THEN  PRINT "No auto
start": GO TO 610
  270 PRINT "Line autostart_____";start
  280 GO TO 610
```

Lines 239-280 Print details of the header for a Basic program file.


48K ZX SPECTRUM
The Knights Quest
PHARAOHS TOMB
48K ZX Spectrum
THE FOREST
LOONY ZOO
DO NOT FEED THE EARTHMAN
COLDITZ
48K ZX Spectrum

AGF

PROGRAMMABLE JOYSTICK INTERFACE for Spectrum or ZX81

ONLY 26·95 +£1·00pp

MICRODRIVE COMPATIBLE

Works with QUICKSHOT II & TRACKBALL

Recognised as the only true Hardware Programmed joystick interface this product offers all the features associated with such a design.

You can use *any* Atari-compatible joystick controller with *any* software for your Sinclair Spectrum or ZX81, not just those with a joystick option.

Movement of the joystick is recognised by the computer *exactly* the same as pressing the appropriate control keys, and can therefore give the most immediate response to that movement. The hardware programmed design works with *all* possible key-reading methods, both BASIC and Machine Code.

Eight directional movement, with or without the fire button being pressed, can be achieved by only programming the left, right, up, down and fire keys required by the game.

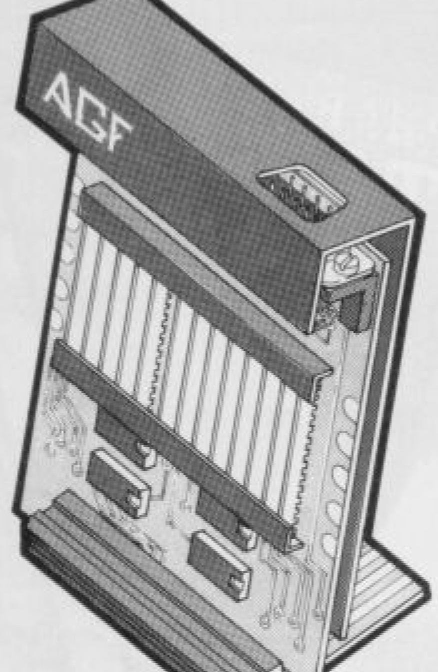
Programming is achieved by a two-digit code, which is looked up on the Programming Chart supplied, for each direction and firing button. These two numbers are then selected on a pair of leads which are clipped onto appropriately numbered strips on the interface.

Once configured this can be marked onto a Quick Reference Programming Card for storing with the game. As the programming is *not* power dependent the interface can be immediately used when next switched on.

The keyboard remains fully functional and can be used simultaneously with the joystick.

An integral rear expansion connector means there is no need to remove the interface to connect other peripherals.

NB. A recent design improvement now means that the AGF Programmable Interface works with the new Quickshot II rapid "Auto Fire" feature.

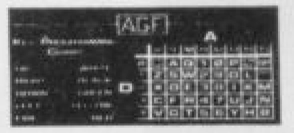


KEY FEATURES

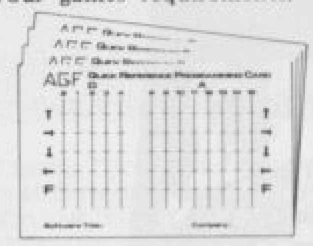
- ★ Programmable design gives TOTAL software support.
- ★ Accepts Atari, Competition Pro, Wico, Starfighter, Quick Shot, Le Stick etc.
- ★ Rear extension connector for all other add-ons.
- ★ Free demo program and instructions.

PACK CONTENTS SUPPLIED

- Programmable Interface Module as illustrated, complete with clip-on programming leads.
- Self adhesive programming chart detailing how to define which key is simulated by UP, DOWN, LEFT, RIGHT, and FIRE.



- One pack of ten Quick Reference Programming Cards for at-a-glance setting to your games requirements.



- 12 months guarantee and full written instructions.

ONLY 11.95 +£1 P&P



Quickshot II® JOYSTICK

NEW IMPROVED GRIP : BUILT-IN STABILIZING SUCTION CUPS

TRIGGER FIRE BUTTON : RAPID AUTO FIRE SWITCH : TOP FIRE BUTTON

AGF Hardware, Bognor Regis, West Sussex PO22 9BY. Telephone: (0243) 823337.

Please allow up to 28 days for delivery.

FROM: MR/MRS/MISS

ADDRESS

SEND C.W.O. (NO STAMP NEEDED) TO: A.G.F. HARDWARE, DEPT. **YS**,

FREEPOST, BOGNOR REGIS, WEST SUSSEX, PO22 9BR.

QTY	ITEM	ITEM PRICE	TOTAL
	PROGRAMMABLE INTERFACE	27.95	
	JOYSTICK(S)	12.95	
	PACK(S) QUICK REFERENCE CARDS	1.00	
	ZX81 ☐ ZX SPECTRUM ☐ Please tick	FINAL TOTAL	

DEALER ENQUIRIES WELCOME *EXPORT PRICES ON APPLICATION*

DO IT WITH TRANS-EXPRESS

TAPE TO MICRODRIVE

MICRODRIVE TO TAPE

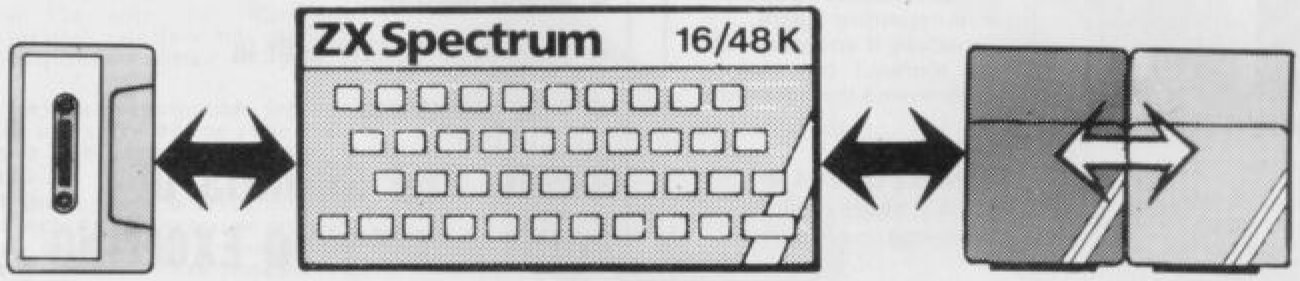
Would you like to transfer your Spectrum programs from

① **MICRODRIVE TO MICRODRIVE** ③ **TAPE TO TAPE**

② **TAPE TO MICRODRIVE** ④ **MICRODRIVE TO TAPE**

Then do it with

TRANS-EXPRESS



TRANS-EXPRESS is the most comprehensive package of four m/c utilities for transferring Spectrum programs. They are user-friendly, simple to use, reliable & very efficient. They go much beyond where similar attempts failed, enabling you to transfer any kind of programs up to the full 48.0K length - provided you do not infringe copyright. **TRANS-EXPRESS** is an essential microdrive companion and an invaluable software back-up utility.

We are offerring the entire package for **£ 9.95** or a combination of TAPE TO MICRODRIVE & MICRODRIVE TO MICRODRIVE for **£ 7.50**. You can also buy each of the four programs separately for **£ 5.50** only.

TAPE TO TAPE

Please send me a copy of **TRANS-EXPRESS** ① ② ③ ④ **Please tick where applicable**

I enclose cheque/PO for ....

(Please add £1.00 for orders outside UK)

Name .... YS

Address ....

....

ROMANTIC ROBOT 113 Melrose Ave, London NW2

```
399 REM not basic program
400 LET t$="code block"
410 IF type=1 THEN  LET t$="numerical a
rray"
420 IF type=2 THEN  LET t$="character a
rray"
430 PRINT "Program type_____";t$: PRINT

440 IF type=3 THEN  PRINT "Destination
address_____";start: PRINT : REM code blo
ck
450 PRINT "Length_______";len;" bytes"
460 RETURN
```

Lines 399-460 The part of the program which deals with the case of the header not being for a Basic program file. t$ is set to equal the file type. If it is a code file, line 440 prints the start address.

```
600 PRINT  FLASH 1;AT 10,6;"THIS WAS NO
T A HEADER"
610 PRINT  FLASH 0;AT 20,1;"Do you want
another go?(y/n)"
620 PAUSE 0: IF INKEY$="y" THEN  GO TO
100
630 STOP
```

Lines 600-630 Lines 120 and 150 checked that the data read in was header. And if it wasn't, then this routine prints the 'NOT A HEADER' message and the user is asked if another header is to be read in.

NOUGHTS & CROSSES

By Mark Burton (48K)

Remember how frustrating it was getting endless draws playing the conventional three-by-three classroom game of Noughts and Crosses? Well, in this program for the 48K Spectrum, you'll be lucky to get a draw, let alone beat the system.

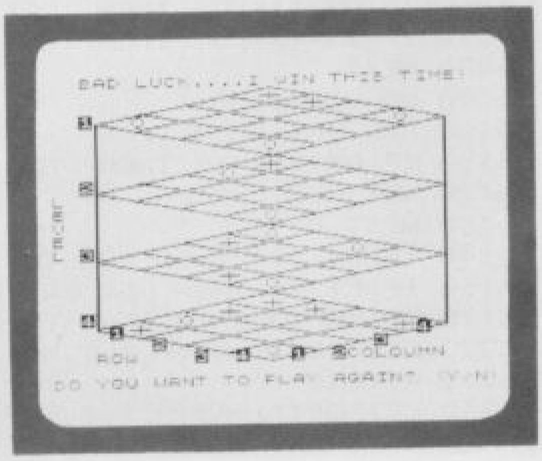
When you first run the program, a message pops up across the top of the screen offering you the option of checking out the instructions — always a useful feature for the first time user. The rules are very easy to follow; you simply input numbers relating to the row, level and column (in that order) and press the Enter key.

Once you've inwardly digested the instructions, pressing any key will thrust you into the action. The four playing grids will be drawn on-screen, each comprising of four-by-four squares, one on top of the other, and each with a numbered label to mark its co-ordinates. After a pause of about 10 seconds while the computer sets up the necessary variables and flags, you'll be asked whether or not you want to start first. Decision made, it's time to begin.

The principle of the game is as old as the hills, so explanation of play will be kept to a minimum. All you have to do is to get four 'X' characters in a row, be they in a horizontal, vertical or diagonal row. But keep an eye on the cunning Speccy to make sure it doesn't get four 'O' characters in a row first.

Once a row of four similar characters has been achieved, a message of congratulations or otherwise will be displayed on-screen. You'll then be asked if you'd like another crack at it — remember, revenge is sweet! Should you input too many characters or make some other silly error, not to worry — the listing is well error-trapped.

And that's it really, no less frustrating than the original — just a bit more challenging.




```
1 REM 4X4X4 NOUGHTS & CROSSES
3 PRINT AT 0,0;"DO YOU WANT INSTRUCTI
ONS?": IF INKEY$="" THEN  GO TO 3
4 IF INKEY$="Y" OR INKEY$="y" THEN  G
O SUB 9600
```

Lines 1-4 Ask if instructions for playing the game are required. If so, the program jumps to the 'instruction' routine at line 960.

```
5 CLS : PLOT 25,146: DRAW 0,-125: DRA
W 100,-21: DRAW 100,21: DRAW 0,125: DRAW
 -200,-42: DRAW 200,-41: DRAW -200,-41:
PLOT 225,21: DRAW -200,42: DRAW 200,42:
DRAW -200,41: DRAW 100,21: DRAW 100,-21
6 FOR a=6 TO 132 STEP 42: PLOT 100,a:
 DRAW 100,20: PLOT 150,a: DRAW -100,20:
NEXT a
7 FOR a=11 TO 137 STEP 42: PLOT 75,a:
 DRAW 100,20: PLOT 175,a: DRAW -100,20:
NEXT a
8 FOR a=16 TO 142 STEP 42: PLOT 50,a:
 DRAW 100,20: PLOT 200,a: DRAW -100,20:
NEXT a
9 PLOT 26,21: DRAW 0,125: PLOT 224,21
: DRAW 0,125: PRINT AT 3,2;"1";AT 8,2;"2
";AT 13,2;"3";AT 18,2;"4";AT 9,0;"L";AT
10,0;"E";AT 11,0;"V";AT 12,0;"E";AT 13,0
;"L";AT 21,3;"ROW";AT 21,21;"COLUMN";AT
19,4;"1";AT 20,7;"2";AT 21,10;"3";AT 21,
13;"4";AT 21,17;"1";AT 21,20;"2";AT 20,2
3;"3";AT 19,26;"4"
```

Lines 5-9 Draw and label the four-by-four-by-four playing grid.

```
10 DIM A$(4,4,4,10): LET CT=0: LET H$=
"O": LET J=0: LET I=0: DIM F$(10): DIM Q
$(4): LET W1=35: LET W2=35: LET W3=25: L
ET W4=30: LET W5=15: LET W6=8: LET W7=15
: LET W8=2
```

Line 10 Sets up the necessary variables and flags. Array A$ holds the current state of the board.

```
20 FOR X=1 TO 4: FOR Y=1 TO 4: FOR Z=1
 TO 4
30 LET F$(1)=CHR$ (X+47)
40 LET F$(2)=CHR$ (Y+47)
50 LET F$(3)=CHR$ (Z+47)
60 LET F$(4 TO 10)="1000000"
70 IF Y=Z THEN  LET F$(7)="1"
80 IF Y+Z=5 THEN  LET F$(7)="2"
90 IF X=Z THEN  LET F$(8)="1"
100 IF X+Z=5 THEN  LET F$(8)="2"
110 IF X=Y THEN  LET F$(9)="1"
120 IF X+Y=5 THEN  LET F$(9)="2"
130 IF F$(7)="1" AND F$(8)="1" THEN  LE
T F$(10)="1"
140 IF F$(9)="1" AND F$(8)="2" THEN  LE
T F$(10)="2"
150 IF F$(8)="1" AND F$(7)="2" THEN  LE
T F$(10)="3"
160 IF F$(8)="2" AND F$(9)="2" THEN  LE
T F$(10)="4"
170 LET V=3
180 IF F$(7)<>"0" THEN  LET V=V+1
190 IF F$(8)<>"0" THEN  LET V=V+1
200 IF F$(9)<>"0" THEN  LET V=V+1
210 IF F$(10)<>"0" THEN  LET V=V+1
220 IF V>=5 THEN  LET V=5
230 LET F$(5)=CHR$ (V*W6)
240 LET A$(X,Y,Z)=F$
250 NEXT Z: NEXT Y: NEXT X
```

Lines 20-250 Set up the array F$ to contain the 'move priority' table and place the current state of play into A$.

```
 260 PRINT AT 0,0;"DO YOU WANT TO PLAY F
IRST? (Y/N)"
 270 IF INKEY$="" THEN  GO TO 270
 280 LET C$=INKEY$
 290 IF C$="Y" OR c$="y" THEN  PRINT AT
0,0;"                              ":
GO TO 1000
 300 IF C$="N" OR c$="n" THEN  PRINT AT
0,0;"                              ":
GO TO 1200
 310 GO TO 270
```

Lines 260-310 Check if the user wants to have the first move. If so, the program jumps to line 1000; otherwise, it jumps to line 1200.

```
 400 IF A$(C,D,E,4)="1" THEN  GO TO 1090
 410 PRINT #0;"      SPACE ALREADY OCCUPI
ED      "
 420 FOR N=1 TO 200: NEXT N
 430 GO TO 1000
```

Lines 400-430 Check if the space you wish to place a character is already occupied. If it is, a message is printed and the program jumps to the 'input move' routine at line 1000.

```
1000 LET G$="0"
1010 INPUT "MOVE: LEVEL ROW COLUMN ";B$
1020 IF LEN B$<>3 THEN  GO TO 1010
1050 FOR N=1 TO 3: IF B$(N)<"1" OR B$(N)
>"4" THEN  GO TO 1010
1060 NEXT N
1070 LET C=VAL B$(1): LET D=VAL B$(2): L
ET E=VAL B$(3)
1080 GO TO 400
```

Lines 1000-1080 Get the player's move and check that the space is not already occupied (via the routine at lines 400-430). The co-ordinates are input in order (level, row and column) and are assigned to the variables C, D and E respectively.

```
1090 LET A$(C,D,E,4)="2"
1100 LET A$(C,D,E,5)=" "
1110 LET K$="X"
1120 GO SUB 9000
1130 GO TO 6000
```

Lines 1090-1130 Place the player's move into the array A$. K$ is the flag used to check whether it's the player's turn to move or the computer's.

```
1140 PRINT AT 0,0;"CONGRATULATIONS....YO
U HAVE WON!"
1150 PRINT #0;"DO YOU WANT TO PLAY AGAIN
? (Y/N)"
1154 LET a=-20: LET b=1
1155 LET a=a+b: BEEP .01,a
1160 IF INKEY$<>"" THEN  GO TO 1170
1165 IF a=20 THEN  LET b=-1: GO TO 1155
1166 IF a=-20 THEN  LET b=1: GO TO 1155
1167 GO TO 1155
1170 LET C$=INKEY$
1180 IF c$="Y" OR c$="y" THEN  RUN
1190 IF c$="N" OR c$="n" THEN  GO TO 119
0
1195 GO TO 1160
```

Lines 1140-1195 The player has managed to put one over on the Speccy. You are given a message of congratulation and asked if you'd like to play another game. Mood music accompanies this momentous decision.

```
1200 IF I>0 THEN  GO TO 1500
1210 IF J>0 THEN  GO TO 1700
1220 LET HIWT=0
1230 FOR X=1 TO 4: FOR Y=1 TO 4: FOR Z=1
 TO 4
1240 IF A$(X,Y,Z,4)<>"1" THEN  GO TO 129
0
1250 LET WT=CODE A$(X,Y,Z,5)
1260 IF WT<HIWT THEN  GO TO 1290
1270 IF WT=HIWT AND RND>.33 THEN  GO TO
1290
1280 LET HIWT=WT: LET C=X: LET D=Y: LET
E=Z
1290 NEXT Z: NEXT Y: NEXT X
1300 LET A$(C,D,E,4)="3": LET A$(C,D,E,5
)=" "
1310 LET G$="1"
1320 IF CT>63 THEN  GO TO 1800
1330 LET K$="O"
1340 GO SUB 9000
1350 GO TO 6000
1500 FOR X=1 TO 4: FOR Y=1 TO 4: FOR Z=1
 TO 4
1510 IF A$(X,Y,Z,6)<"1" OR A$(X,Y,Z,6)>"
7" THEN  GO TO 1590
1520 IF A$(X,Y,Z,4)<>"1" THEN  GO TO 159
0
1530 LET C=X: LET D=Y: LET E=Z: LET P=VA
L A$(X,Y,Z,6)*20
1540 LET K$="O"
1550 GO SUB 9000
1570 PRINT AT 0,0;"  BAD LUCK....I WIN T
HIS TIME!"
1580 GO TO 1150
1590 NEXT Z: NEXT Y: NEXT X
1600 LET I=0
1610 GO TO 1210
1700 FOR X=1 TO 4: FOR Y=1 TO 4: FOR Z=1
 TO 4
1710 IF A$(X,Y,Z,6)<>"8" THEN  GO TO 175
0
1720 IF A$(X,Y,Z,4)<>"1" THEN  GO TO 175
0
1730 LET C=X: LET D=Y: LET E=Z: LET A$(X
,Y,Z,6)="0": LET J=J-1
1740 GO TO 1300
1750 NEXT Z: NEXT Y: NEXT X
1760 LET J=0
1770 GO TO 1220
```

Lines 1200-1770 Hold the logic which checks whether someone has won the game. If it's the computer, a message offering you commiserations appears. The program then jumps to the routine asking if you'd like another game.

```
1800 PRINT AT 0,0;"WELL THE GAME SEEMS T
O BE A DRAW"
1810 GO TO 1150
```

Lines 1800-1810 If you manage to hold off the Speccy to a draw, these lines give you a relevant on-screen message and jump to the routine at line 1150 offering you another chance for your revenge.

```
6000 LET CT=CT+1
6010 FOR P=20 TO 140 STEP 20
6020 LET L=0: LET M=0
6040 FOR T=1 TO 4
6050 GO SUB (6200+P)
6060 IF Q$(T)="2" THEN  LET L=L+1
6070 IF Q$(T)="3" THEN  LET M=M+1
6080 NEXT T
6090 GO SUB 6600
6100 FOR T=1 TO 4
6110 IF Q$(T)="1" THEN  GO SUB (6400+P)
6120 NEXT T
6130 NEXT P
6140 IF G$="0" THEN  GO TO 1200
6150 IF CT>63 THEN  GO TO 1800
6170 GO TO 1000
6220 LET Q$(T)=A$(T,D,E,4)
6230 RETURN
6240 LET Q$(T)=A$(C,T,E,4)
```

```
6250 RETURN
6260 LET Q$(T)=A$(C,D,T,4)
6270 RETURN
6280 IF A$(C,D,E,7)="1" THEN  LET Q$(T)=
A$(C,T,T,4)
6285 IF A$(C,D,E,7)="2" THEN  LET Q$(T)=
A$(C,T,5-T,4)
6290 RETURN
6300 IF A$(C,D,E,8)="1" THEN  LET Q$(T)=
A$(T,D,T,4)
6305 IF A$(C,D,E,8)="2" THEN  LET Q$(T)=
A$(T,D,5-T,4)
6310 RETURN
6320 IF A$(C,D,E,9)="1" THEN  LET Q$(T)=
A$(T,T,E,4)
6325 IF A$(C,D,E,9)="2" THEN  LET Q$(T)=
A$(T,5-T,E,4)
6330 RETURN
6340 LET R=VAL A$(C,D,E,10)
6341 LET Q$(T)="1"
6342 IF R=0 THEN  RETURN
6345 GO TO (6200+P+R*10)
6350 LET Q$(T)=A$(T,T,T,4)
6355 RETURN
6360 LET Q$(T)=A$(T,T,5-T,4)
6365 RETURN
6370 LET Q$(T)=A$(T,5-T,T,4)
6375 RETURN
6380 LET Q$(T)=A$(T,5-T,5-T,4)
6385 RETURN
6420 LET A$(T,D,E,5)=CHR$ (CODE A$(T,D,E
,5)+WW)
6430 RETURN
6440 LET A$(C,T,E,5)=CHR$ (CODE A$(C,T,E
,5)+WW)
6450 RETURN
6460 LET A$(C,D,T,5)=CHR$ (CODE A$(C,D,T
,5)+WW)
6470 RETURN
6480 IF A$(C,D,E,7)="1" THEN  LET A$(C,T
,T,5)=CHR$ (CODE A$(C,T,T,5)+WW)
6485 IF Q$(T)="1" AND A$(C,D,E,7)="2" TH
EN  LET A$(C,T,5-T,5)=CHR$ (CODE A$(C,T,
5-T,5)+WW)
6490 RETURN
6500 IF A$(C,D,E,8)="1" THEN  LET A$(T,D
,T,5)=CHR$ (CODE A$(T,D,T,5)+WW)
6505 IF A$(C,D,E,8)="2" THEN  LET A$(T,D
,5-T,5)=CHR$ (CODE A$(T,D,5-T,5)+WW)
6510 RETURN
6520 IF A$(C,D,E,9)="1" THEN  LET A$(T,T
,E,5)=CHR$ (CODE A$(T,T,E,5)+WW)
6525 IF A$(C,D,E,9)="2" THEN  LET A$(T,5
-T,E,5)=CHR$ (CODE A$(T,5-T,E,5)+WW)
6530 RETURN
6540 IF R<>0 THEN  GO TO (6400+P+R*10)
6545 RETURN
6550 LET A$(T,T,T,5)=CHR$ (CODE A$(T,T,T
,5)+WW)
6555 RETURN
6560 LET A$(T,T,5-T,5)=CHR$ (CODE A$(T,T
,5-T,5)+WW)
6565 RETURN
6570 LET A$(T,5-T,T,5)=CHR$ (CODE A$(T,5
-T,T,5)+WW)
6575 RETURN
6580 LET A$(T,5-T,5-T,5)=CHR$ (CODE A$(T
,5-T,5-T,5)+WW)
6585 RETURN
6610 IF G$="1" THEN  GO TO 6710
6625 LET WW=0
6630 IF L=4 THEN  GO TO 7000
6635 IF M<>0 THEN  GO TO 6675
6640 IF L=3 THEN  LET WW=(W2-W4): GO TO
7050
6660 IF L=2 THEN  LET WW=(W4-W7)
6670 IF L=1 THEN  LET WW=(W7-W6)
6675 IF L<>1 THEN  RETURN
6680 IF M=1 THEN  LET WW=(W8-W5)
6690 IF M=2 THEN  LET WW=(W8-W3)
6700 RETURN
6710 LET WW=0
6720 IF M=4 THEN  GO TO 1540
6725 IF L<>0 THEN  GO TO 6765
6730 IF M=3 THEN  LET WW=(W1-W3): GO TO
7100
6750 IF M=2 THEN  LET WW=(W3-W5)
6760 IF M=1 THEN  LET WW=(W5-W6)
6765 IF M<>1 THEN  RETURN
6770 IF L=1 THEN  LET WW=(W8-W7)
6780 IF L=2 THEN  LET WW=(W8-W3)
6800 RETURN
```

Lines 6000-6800 Hold the logic to instruct the Speccy in the finer arts of Noughts and Crosses. You'll find it a mean opponent!

```
7000 LET K$="X"
7010 GO SUB 7150
7020 GO SUB 9000
7030 GO TO 1140
7050 LET H$="8": LET J=J+1
7060 GO TO 7200
7100 LET H$=STR$ (P/20)
7110 LET I=1: GO TO 7200
7150 FOR T=1 TO 4
7160 LET H$="9"
7170 GO SUB (7250+P)
7180 NEXT T
7190 RETURN
7200 FOR T=1 TO 4
7210 GO SUB (7250+P)
7220 NEXT T
7230 RETURN
7270 LET A$(T,D,E,6)=H$
7280 RETURN
7290 LET A$(C,T,E,6)=H$
7300 RETURN
7310 LET A$(C,D,T,6)=H$
7320 RETURN
7330 IF A$(C,D,E,7)="1" THEN  LET A$(C,T
,T,6)=H$
7335 IF A$(C,D,E,7)="2" THEN  LET A$(C,T
,5-T,6)=H$
7340 RETURN
7350 IF A$(C,D,E,8)="1" THEN  LET A$(T,D
,T,6)=H$
7355 IF A$(C,D,E,8)="2" THEN  LET A$(T,D
,5-T,6)=H$
7360 RETURN
7370 IF A$(C,D,E,9)="1" THEN  LET A$(T,T
,E,6)=H$
7375 IF A$(C,D,E,9)="2" THEN  LET A$(T,5
-T,E,6)=H$
7380 RETURN
7390 LET R=VAL A$(C,D,E,10)
7395 GO TO (7250+P+R*10)
7400 LET A$(T,T,T,6)=H$
7405 RETURN
7410 LET A$(T,T,5-T,6)=H$
7415 RETURN
7420 LET A$(T,5-T,T,6)=H$
7425 RETURN
7430 LET A$(T,5-T,5-T,6)=H$
7435 RETURN
```

Lines 7000-7435 This is the data controlling the player's move.

```
9000 IF E=1 THEN  GO TO 9100
9010 IF E=2 THEN  GO TO 9200
9020 IF E=3 THEN  GO TO 9300
9030 IF E=4 THEN  GO TO 9400
9040 RETURN
9100 IF C=1 THEN  LET XX=132
9110 IF C=2 THEN  LET XX=90
9120 IF C=3 THEN  LET XX=48
```

```
9130 IF C=4 THEN  LET XX=6
9135 LET YY=119
9140 IF D=1 THEN  LET XX=XX+15: LET YY=4
3
9150 IF D=2 THEN  LET XX=XX+10: LET YY=6
9
9160 IF D=3 THEN  LET XX=XX+5: LET YY=94
9170 GO TO 9500
9200 IF C=1 THEN  LET XX=137
9210 IF C=2 THEN  LET XX=95
9220 IF C=3 THEN  LET XX=53
9230 IF C=4 THEN  LET XX=11
9240 LET YY=144
9250 IF D=1 THEN  LET XX=XX+15: LET YY=6
9
9260 IF D=2 THEN  LET XX=XX+10: LET YY=9
4
9270 IF D=3 THEN  LET XX=XX+5: LET YY=11
9
9280 GO TO 9500
9300 IF C=1 THEN  LET XX=142
9310 IF C=2 THEN  LET XX=100
9320 IF C=3 THEN  LET XX=58
9330 IF C=4 THEN  LET XX=16
9340 LET YY=169
9350 IF D=1 THEN  LET XX=XX+15: LET YY=9
4
9360 IF D=2 THEN  LET XX=XX+10: LET YY=1
19
9370 IF D=3 THEN  LET XX=XX+5: LET YY=14
4
9380 GO TO 9500
9400 IF C=1 THEN  LET XX=147
9410 IF C=2 THEN  LET XX=105
9420 IF C=3 THEN  LET XX=63
9430 IF C=4 THEN  LET XX=21
9440 LET YY=194
9450 IF D=1 THEN  LET XX=XX+15: LET YY=1
19
9460 IF D=2 THEN  LET XX=XX+10: LET YY=1
44
9470 IF D=3 THEN  LET XX=XX+5: LET YY=16
9
9480 GO TO 9500
```

Lines 9000-9480 Calculate the plot position for each of the moves made in the game.

```
9500 BEEP .2,29: BEEP .2,25
9505 IF K$="X" THEN  PLOT YY,XX: DRAW 12
,0: PLOT YY+6,XX+3: DRAW 0,-6
9510 IF K$="O" THEN  CIRCLE YY+6,XX,3
9520 GO TO 9040
```

Lines 9500-9520 Plot an 'X' or 'O' in the position calculated by the routine in lines 9000-9480, depending on the value in K$ (ie. whose turn it is).

```
9600 CLS : PRINT "  3D 4X4X4 NOUGHTS AND
 CROSSES  "
9610 PRINT  OVER 1;AT 0,1;"______________
_______________"
9620 PRINT
9630 PRINT "TO ENTER YOUR MOVE TYPE IN A
    THREE DIGIT NUMBER:-
                                1ST DIG
IT - LEVEL              2ND DIGIT - ROW
              3RD DIGIT - COLUMN "
9640 PRINT : PRINT "    (FOLLOWED BY 'ENT
ER')"
9650 PRINT #0;"  PRESS ANY KEY TO START
GAME "
9660 IF INKEY$="" THEN  GO TO 9660
9670 RETURN
```

Lines 9600-9999 Print the instructions for the first-time user. YS

ADVERTISERS INDEX

If the apes don't get you the crocodiles will!



Our intrepid explorer is forced to leap across rivers on stepping-stones, cut his way through trees, swing over a fire-pit and by-pass a flame-throwing dragon ...

... all the time harassed by falling coconuts, rampaging apes and ravenous crocodiles!

JUNGLE TROUBLE –
Another great quality game from the house which brought you
HARRIER ATTACK and SCUBA DIVE



DURELL software

available from

W. H. SMITH • LASKYS • GREENS at DEBENHAMS • BOOTS • MARTINS • COMPUTERS FOR ALL • SPECTRUM CENTRES and many other retail outlets

or order direct from
CASTLE LODGE, CASTLE GREEN,
TAUNTON, SOMERSET. TA1 4AB

MAIL ORDER

All games £6.95 including postage and packing.

TYPE OF COMPUTER: ....
CASSETTE TITLES:
1) ....
2) ....
3) ....
NAME: ....
ADDRESS: ....

Overseas customers please allow £1.00 extra

CIRCE

THE ASSENT OF EVERISS

***Controversy surrounds Imagine Software — love 'em or hate 'em, you can't ignore 'em. Paul Walton rattles some skeletons in the company of* Bruce Everiss.**



Picture a giant office-cum-bedroom infested with 50 computer terminals atop plush desks and scattered higgledly-piggledy around the expensive, carpeted floor. The machines are powerful Sage IVs. It's here games are developed, to be squirted down to the host machines — such as the Spectrum. It's a jungle where Imagine employs close on 100 programmers, technicians, artists and musicians. It's even home of games whizz-kid Eugene Evans.

Bruce Everiss is a 26 year-old smooth, silk-suited, micro-veteran and he's Imagine's operations director. We asked him what's coming along Spectrum-wise. "We've got three more titles on the way over the next month or so. *Cosmic Cruiser*'s a space game and *BC Bill*'s a cutie game about a prehistoric guy who lives in a cave and has to catch his food. The thing that both games have in common is very good graphics and sound."

So what's the best-seller on Spectrum? "Well, it's always the latest title. We've just brought out *Pedro* and it seems to be getting into some people's charts. Overall, I think *Arcadia* is the biggest. By a large margin most are for the Spectrum, but this situation is changing as the market changes."

How about the QL? "When there are lots of QLs out there, obviously it will be natural for us to write for it — because we've got lots of 68000 experience. In fact, we do have products we could adapt for the QL — but we'll have to wait and see. There's no point in letting the cat out of the bag."

Pedro was the first Imagine games cassette to feature a jazzy new inlay card and apparently the presentation's going to get better all the time. "A large number of small things came together at once. They've got screen photographs and a game description. We've gone to fifth colour on the front side (vastly improving the appearance and adding a fourth flap). They've got a programmer profile, company profile and very extensive playing instructions. All sorts of things. Everything that anybody could possibly put together on an insert card — we've done it."

Talk switched to the 'first you see it, then you don't, price reduction to under the £4 barrier. What kind of planning went into such nervous pitching? "We knew that dropping the price would increase sales, but what we hadn't bargained for was the industry reaction — which was universally unfavourable, from both the distributors and other software houses. The feeling was that if we did do it, it would upset the marketplace to such a degree that it would put many smaller software houses out of business.

"What we're concerned with is giving the customers value for money. The average price of a computer game in this country is £7.20, and if you look at *Valhalla* and *Alchemist* it's fairly obvious where the value for money is. On the other hand, it's now possible to write a fairly simple game in a week — which obviously you could sell for £1.99, or whatever. That's what Mastertronic are doing."

So, what about Imagine's next range of games — the Megagames — at around the £30 mark? "Looking at *Bandersnatch*, which is coming out sometime in the Summer, it's already beginning to look like it's going to contain three man-year's work. We've got 17 people working on this project — and at £30, there's going to be so much there that it's still going to be great value."

Anybody in the know at Imagine gets a little cagey when you ask about the so-called Megagames — and even Everiss isn't saying that much more. "The thing about it is that the game is so big and complex and involved — and it contains several new areas, things that have never been done before. We aren't going to release it until it's perfect — the only analogy that we can use without giving the game away is that it's going to make anything that's gone before look like Noughts and Crosses.

"No-one's even seen them yet! They're so secret that most people at Imagine know nothing about them. Even the people who are working on the project only know sufficient to do their own piece of the work — we give them information on a 'need to know' basis. What we're worried about is somebody else finding out what we're doing and emulating it."

That was interesting, because much of Imagine's business now seems fraught with secrecy and intrigue — perhaps it's the price of success. For instance, there's the Marshall Cavendish affair where Imagine was going to produce games to accompany *Input* magazine. Eventually the project was dropped. Everiss counters rumours that they were late and the product wasn't up to scratch. "The idea was that each fortnight it would have a game on it for several machines. But the original concept was that these should be average run-of-the-mill games. As we started developing the games, we put them out to be play-tested — which involves comparing them against the reviewer's favourite game. So the games were enhanced and enhanced and so on, so that in the end they became so good that it wasn't worth our while putting them though Marshall Cavendish."

Imagine has been active in trying to stamp out software piracy. The company mailed out a letter to magazines asking them to be careful not to publicise or advertise any offending material. Did it work? "I think that some of the weeklies are filtering their adverts more thoroughly and the Advertising Execs have acted quite strongly to support our point. The trouble is that we only have so much time and money to put into things — and we can't spend all our time trying to wipe out piracy.

"We've done as much as we can. The Guild of Software Houses won't let us join; if we were in GOSH then obviously we could all work together — sort out common problems. But I think we're too big for them. I think that they want to keep it as a small, mutual back-slapping organisation really."

Finally, there was the question of the new company's logo — is Imagine being renamed? "Ah yes, the Creative Technology Group Ltd. That's the name for the overall organisation that we're intending to put in with Imagine Software. It's the name of a company that Imagine would eventually become. That's the concept — but it's not actually put through yet." YS

GRAFPAD

for Sinclair Spectrum . . . Drawing into the future!

Now you too can have the benefit of drawing your own applications by the simple use of our NEW ZX Spectrum version of Grafpad!

THE GRAFPAD comes complete with Cassette Programme, Special Keyboard, Overlay, Plug, Light Pen and, of course, a comprehensive manual full of tips and instructions . . . all at the usual high quality expected from **BRITISH MICRO!**

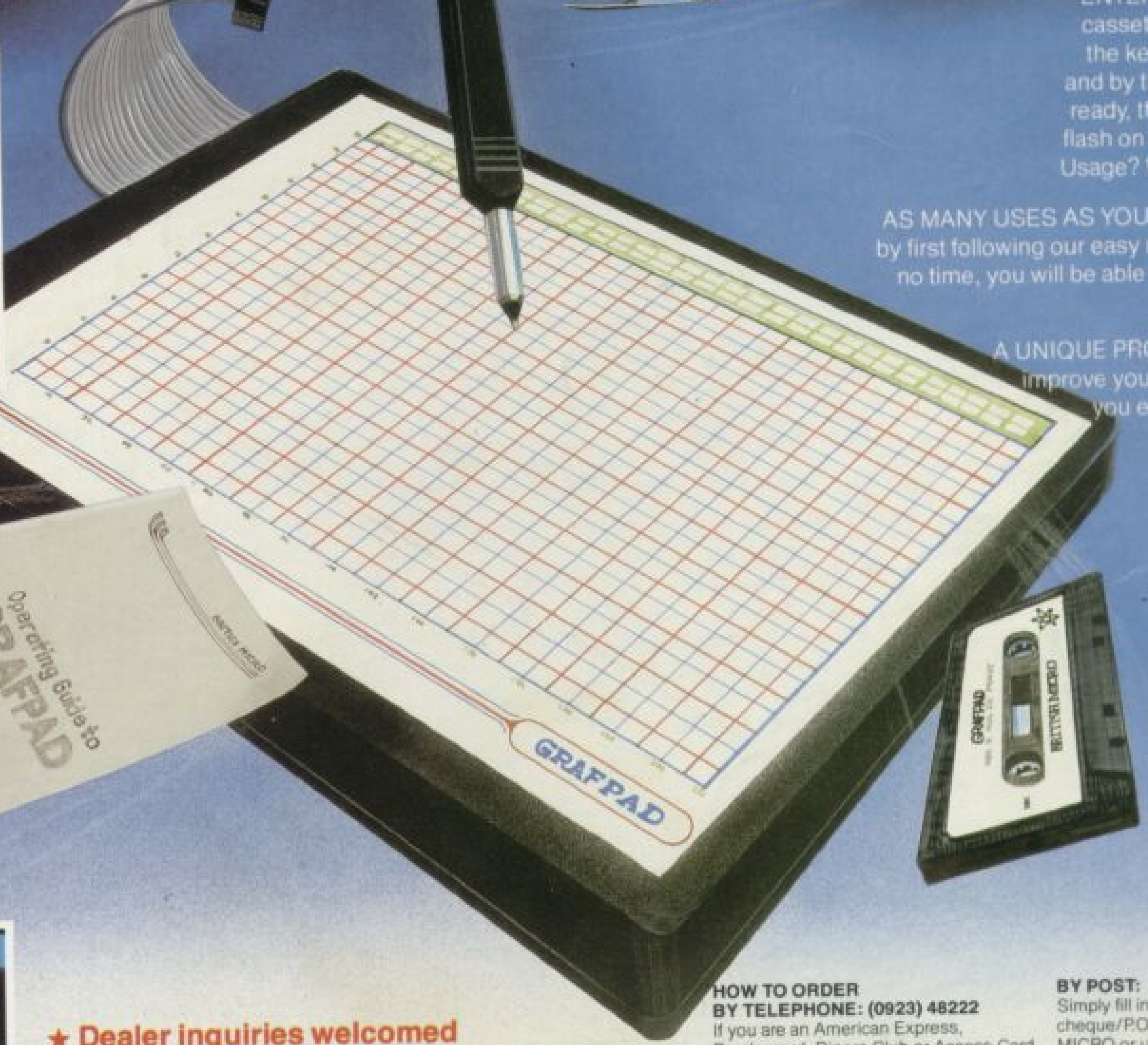




STARTING UP is simple. Connect the Grafpad, switch on your Spectrum and connect the cassette player, type "LOAD" and press ENTER, then start the cassette player, place the keyboard overlay, and by the time you are ready, the "MENU" will flash on to your screen! Usage? Well there is . . .

AS MANY USES AS YOU CAN IMAGINE by first following our easy manual, then, in no time, you will be able to utilise as you Think best!

A UNIQUE PRODUCT that will improve your skills and give you endless hours of excitement!



Size
25mm height
55mm width
260 depth
weight 1.2k (gross)



★ Dealer inquiries welcomed

★ Special discounts to educational authorities & government departments

★ Export inquiries invited

HOW TO ORDER
BY TELEPHONE: (0923) 48222
If you are an American Express, Barclaycard, Diners Club or Access Card Holder simply telephone us giving your Card No., Name, Address and item(s) required and your order will be despatched within 48 hours!

BY POST:
Simply fill in the coupon enclosing your cheque/P.O. made payable to: BRITISH MICRO or use the special section for Credit Card Holders and post to the address below.
OVERSEAS ORDERS:
Please add £10 per item ordered.

BRITISH MICRO

A HEGOTRON GROUP COMPANY

Unit Q2, Penfold Works,
Imperial Way, Watford, HERTS WD2 4YY
TEL: (0923) 48222. TELEX: 946024

★ INQUIRE ABOUT BBC VERSION.

Post to: BRITISH MICRO, UNIT Q2, PENFOLD WORKS, IMPERIAL WAY, WATFORD, HERTS. WD2 4YY.

Please send me GRAFPAD for: SINCLAIR SPECTRUM
☐ BBC MODEL B ☐ COMMODORE 64
If you require details of the above versions, please tick.

Qty.	Item	Ex. VAT	Inc. VAT	Total
	Grafpad Complete	£125.00	£143.75	
			Postage, Packing & Insurance	£5.00
			TOTAL	£

I enclose my cheque for / P.O. £...........

I prefer to pay with my American Express, Barclaycard, Diners Club, Access Card.
(Please delete whichever is not applicable)

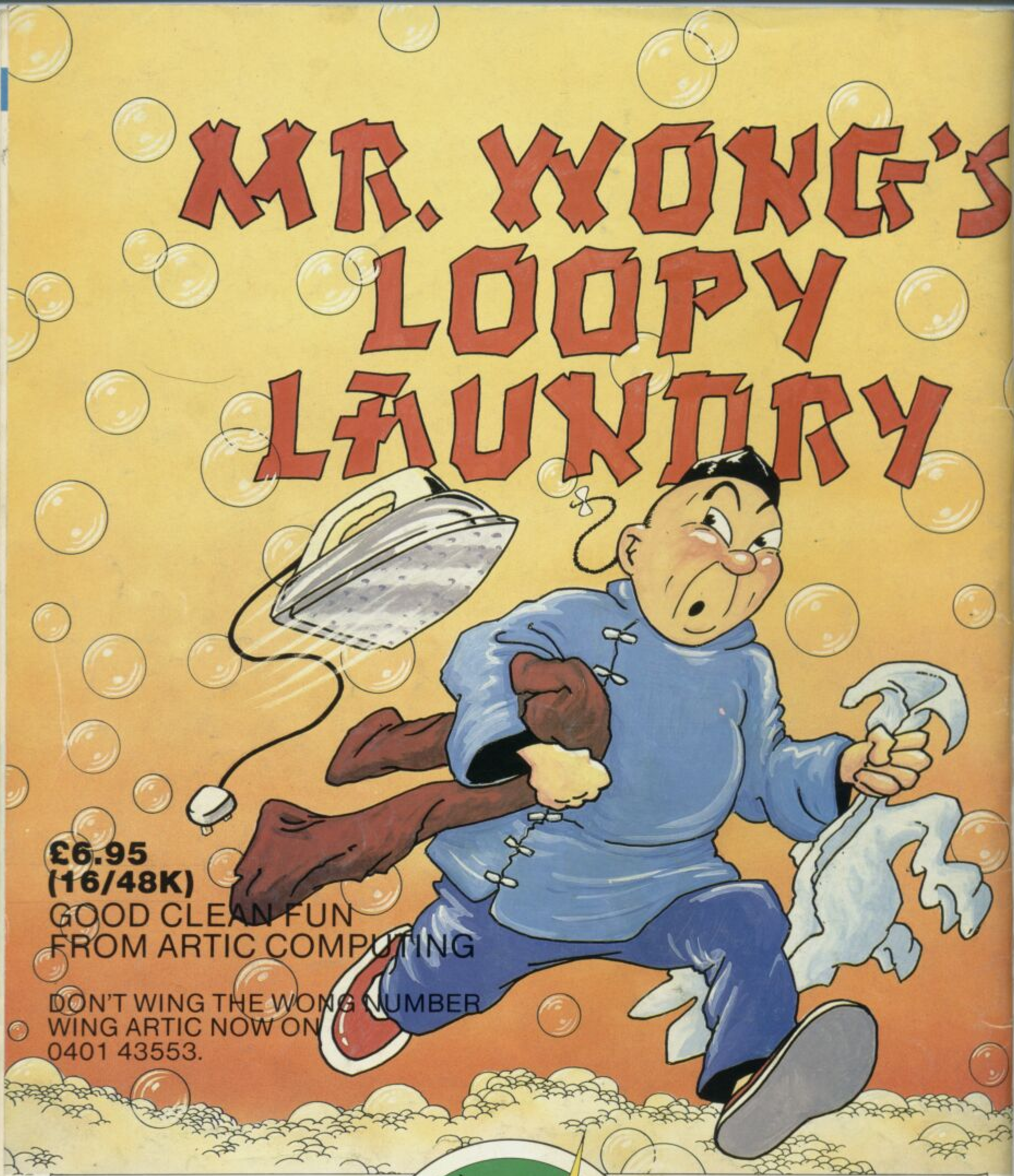
Card No. ....

Signature ....

Name ....

Address ....

Address above must be the same as Card Holder


MR. WONG'S LOOPY LAUNDRY
£6.95
(16/48K)
GOOD CLEAN FUN
FROM ARTIC COMPUTING
DON'T WING THE WONG NUMBER
WING ARTIC NOW ON
0401 43553.